EAS

DigiBox CODI®

Handbook

2a02176

Chyron Corporation
5 Hub Drive
Melville, NY 11747
631-845-2000
www.chyron.com

Copyright 2002 Chyron Corporation

Table of Contents

Section 1 - Installation and Setup 1-1

1.1 An Important Note On Physical Installation of DigiBox CODI 1-1
1.2 Required and Optional Hardware 1-1
1.3 EAS DigiBox CODI Specifications and Diagrams 1-2
1.4 Data Connections 1-6
1.4.1 RS-232/RS-422 Selection 1-6
1.4.2 Making or Repairing Serial Cables 1-7
1.4.3 Serial Cabling To EAS Encoder/Decoder 1-8
1.4.4 Serial Communication Settings 1-9
1.5 Video Monitor Hookup 1-9
1.6 Video Input/Outputs 1-11
1.6.1 Selection of Genlock Source 1-12
1.6.2 Keying Over Live Video 1-12
1.6.3 Video Bypass Relay 1-13
1.7 GPI/O 1-13

Section 2 - Initial Operation 2-1

2.1 Working With EAS DigiBox CODI's Software 2-1
2.1.1 About DigiBox CODI's CompactFlash Cards 2-1
2.1.2 Removing and Replacing the Internal CompactFlash Memory Card 2-2
2.1.3 Working With CompactFlash Memory Cards 2-3
2.1.3.1 Writing To/Reading From CompactFlash Cards 2-3
2.1.3.2 Formatting CompactFlash Cards For Use With DigiBox CODI 2-3
2.1.4 Editing EAS DigiBox CODI's Files 2-4
2.1.4.1 CODIEAS.mi And CODIDISP.txt 2-4
2.1.4.2 Setting Serial Communication Speed by Editing CODIINIT.txt 2-5
2.2 Operation 2-6

Section 3 - DigiBox CODI Commands 3-1

3.1 INTRODUCTION 3-1
3.2 DigiBox CODI Command Features 3-2
3.2.1 Commands Not Used 3-3
3.2.2 Command Notation 3-4

3.3 Background Commands 3-5
3.3.1 Download Background (HSI) Mode 1, 7 3-6
3.3.2 Download Background (RGB) Mode 1, 7.................... 3-7
3.3.3 Download Background (RGB) Mode 2, 8.................... 3-8
3.3.4 Download Background (HSI) Mode 2, 8 3-9
3.3.5 Download Background (File-Blend) - Mode 3 3-10
3.3.6 Download Background (File-CUT) - Mode 4.................... 3-11
3.3.7 Download Background (PCI-Blend) - Modes 5, 6, and 7.................... 3-12
3.3.8 Download Background (PCI-CUT) - Modes 8, 9, 10, 11.................... 3-13
3.3.9 Upload Background (HSI) Mode 2 3-14
3.3.10 Background Erase 3-15
3.3.11 Set Scanline of Background Buffer.................... 3-16
3.4 Color Commands 3-17
3.4.1 Color Restore 3-18
3.4.2 Color Select 3-19
3.4.3 Color Map 3-20
3.4.4 Color Transparency 3-21
3.4.5 Upload Transparency.................... 3-22
3.5 Display Commands 3-23
3.5.1 Upload Display.................... 3-24
3.5.2 Fade Rectangle.................... 3-25
3.5.3 Dissolve Rectangle 3-26
3.5.4 Bit Blitting - Function Call 3-27
3.6 Event Timer Commands 3-29
3.6.1 Event Timer Index.................... 3-30
3.6.2 Event Timer Display.................... 3-31
3.6.3 Event Timer Setup.................... 3-32
3.6.4 Event Timer Begin.................... 3-34
3.6.5 Event Timer Laps 3-35
3.6.6 Event Timer Erase.................... 3-36
3.6.7 Event Timer Time.................... 3-37
3.6.8 Event Timer Color.................... 3-38
3.6.9 Event Timer Font.................... 3-39
3.6.10 Event Timer Position.................... 3-40
3.7 File System Commands 3-41
3.7.1 Set Message Directory Path 3-42
3.7.2 Set Font Directory Path 3-43
3.7.3 Set Graphic Directory Path 3-44
3.8 Flash Commands....................3-45
3.8.1 Flash Setup 3-46
3.8.2 Flash Begin 3-47
3.8.3 Flash Multiple 3-48
3.8.4 Flash End.................... 3-49

3.9 Font Commands 3-50
3.9.1 Download Font 3-51
3.9.2 Download iNFiNiT! Font 3-52
3.9.3 Font Select 3-53
3.9.4 System Font 3-54
3.9.5 Download Virtual Font 3-55
3.10 GPIO Commands 3-56
3.10.1 GPI Setup 3-57
3.10.2 GPI Begin 3-58
3.10.3 GPI Time 3-59
3.10.4 GPI Time Reset 3-60
3.10.5 GPI Output 3-61
3.10.6 GPI Input 3-62
3.10.7 Register GPIO Callback Procedure 3-63
3.11 Graphic Commands 3-64
3.11.1 Download Graphic 3-65
3.11.2 Download Pixels 3-67
3.11.3 Graphic Erase 3-68
3.11.4 Graphic Transparency 3-69
3.11.5 Upload Pixels 3-70
3.12 Letter Commands 3-71
3.12.1 Letter Blank 3-72
3.12.2 Letter Up 3-73
3.12.3 Letter Down 3-74
3.12.4 Letter Left 3-75
3.12.5 Letter Right 3-76
3.12.6 Hexadecimal UniCode Characters 3-77
3.13 Line Draw Commands 3-78
3.13.1 Draw Color 3-79
3.13.2 Draw Edge 3-80
3.13.3 Draw Line (To) 3-81
3.13.4 Draw Move (To) 3-82
3.13.5 Draw Size 3-83
3.14 Message Commands 3-84
3.14.1 Download Message 3-85
3.14.2 Download iNFiNiT! Message 3-86
3.14.3 Download iNFiNiT! Message in Non-Display Mode 3-87
3.14.4 Message Display 3-88
3.14.5 Message Control 3-89
3.14.6 Upload Message 3-90
3.14.7 Message Number 3-93

3.15 Message Effect Commands 3-94
3.15.1 Message Effect Cover 3-95
3.15.2 Message Effect Hide 3-96
3.15.3 Message Effect Peel 3-97
3.15.4 Message Effect Pour 3-98
3.15.5 Message Effect Reveal 3-99
3.15.6 Message Effect Split 3-100
3.15.7 Message Effect Weave 3-101
3.15.8 Message Effect Crawl Setup 3-102
3.15.9 Message Effect Crawl Begin 3-103
3.15.10 Message Effect Fade 3-105
3.15.11 Message Effect Dissolve 3-106
3.15.12 Message Effect Matrix Wipe 3-107
3.15.13 Message Effect Push 3-109
3.15.14 Message Effect Roll Setup 3-110
3.15.15 Message Effect Roll Begin 3-111
3.15.16 Message Effect Roll Erase 3-112
3.15.17 Message Effect - iNFiNiT! Messages 3-113
3.16 Palette Commands 3-114
3.16.1 Download Palette (HSI) 3-115
3.16.2 Download Palette (RGB) 3-116
3.16.3 Upload Palette 3-117
3.16.4 Download iNFiNiT! Palette 3-118
3.17 Page Commands 3-119
3.17.1 Page Blank 3-120
3.17.2 Page Center 3-121
3.17.3 Page Erase 3-122
3.17.4 Page Up 3-123
3.17.5 Page Down 3-124
3.17.6 Page Left 3-125
3.17.7 Page Right 3-126
3.18 Row Commands 3-127
3.18.1 Row Swap 3-128
3.18.2 Row Color 3-129
3.18.3 Row Font 3-130
3.18.4 Row Blank 3-131
3.18.5 Row Center 3-132
3.18.6 Row Up 3-133
3.18.7 Row Down 3-134
3.18.8 Row Left 3-135
3.18.9 Row Right 3-136
3.18.10 Row Insert 3-137
3.18.11 Row Expand 3-138
3.18.12 Row Squeeze 3-139

3.19 System Commands 3-140
3.19.1 System Acknowledge 3-141
3.19.2 System Cursor (Prompt Channel) 3-142
3.19.3 System Display 3-143
3.19.4 System Execute 3-144
3.19.5 System Home 3-145
3.19.6 System Kern 3-146
3.19.7 System Lock 3-147
3.19.8 System Communication Mode 3-148
3.19.9 System Overwrite/Insert 3-150
3.19.10 System DigiBox CODI Patterns 3-151
3.19.11 System Reset 3-152
3.19.12 System Safe Title 3-153
3.19.13 System Update 3-155
3.19.14 System Wait 3-156
3.19.15 Set 525 NTSC Display Mode 3-157
3.19.16 Set 625 PAL Display Mode 3-158
3.19.17 Display ON/OFF 3-159
3.20 Tab Commands 3-160
3.20.1 Download Tabs 3-161
3.20.2 Tab Update No Display 3-162
3.20.3 Upload Tabs 3-163
3.20.4 Tab Position 3-164
3.20.5 Tab Update 3-165
3.20.6 Tab Erase 3-166
3.20.7 Tab Typewriter 3-167
3.20.8 Highlight Tabs 3-168
3.21 Video Commands 3-169
3.21.1 Video Transparency 3-170
3.21.2 Video Window 3-171
3.21.3 Video Blank 3-172
3.21.4 Video Insert 3-173
3.21.5 Video Only 3-174
3.21.6 Video Verify 3-175
3.21.7 Video Priority Set 3-176
3.21.8 Video Exchange 3-177
3.21.9 Force Video Key IN “OFF” 3-178
3.21.10 Horizontal Video Adjustment 3-179
3.21.11 Vertical Video Adjustment 3-180
3.21.12 Video Genlock 3-181
3.21.13 Video Capture Setup 3-182
3.21.14 Video Capture Begin 3-183
3.21.15 Video Capture Display 3-184
3.21.16 Video Preview Display 3-185

3.22 Watch Commands 3-186
3.22.1 Watch Time 3-187
3.22.2 Watch Setup 3-188
3.22.3 Watch Color 3-189
3.22.4 Watch Font 3-190
3.22.5 Watch Position 3-191
3.22.6 Watch Display 3-192
3.22.7 Watch Erase 3-193
3.22.8 Watch Index 3-194
3.23 Miscellaneous Commands 3-195
3.23.1 Boot pcCODI 3-196
3.23.2 Close pcCODI 3-197
3.23.3 Build Codi Font - Function Call 3-198
3.23.4 Get pcCODI Information - Function Call 3-201
3.23.5 Get Number of DigiBox CODI Boards - Function Call 3-202
3.23.6 Command Capture On/Off 3-203
3.23.7 Convert Chyron RGBA File to DIB - Function Call 3-204
3.23.8 Minimize TriMedia Console Window - Function Call 3-205

Section 1: Installation and Setup

This section contains instructions for installing and setting up EAS DigiBox CODI. Figure 1-2 provides a general overview of how the connections are made.

1.1 An Important Note On Physical Installation of DigiBox CODI

DigiBox CODI is designed for rack-mounting. HOWEVER, **rack slides** must NOT be used to mount the unit!! Essential venting takes place at the sides of the chassis and the vents should not be blocked in any way.

Use the mounting holes at the corners of the front panel, as shown below.

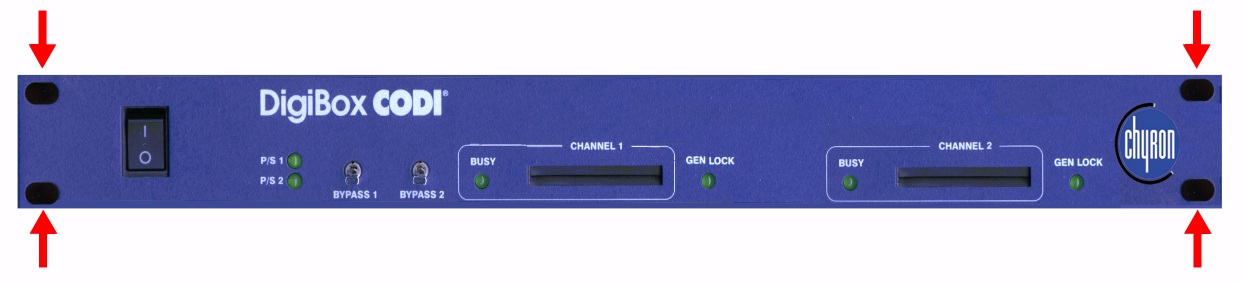


Figure 1-1. DigiBox CODI Front Panel

Before rack-mounting DigiBox CODI, it is recommended that you refer to §1.4.1; there is a switch on the bottom of the unit that should be set before mounting.

1.2 Required and Optional Hardware

- An FCC Part 11-compliant EAS receiver
- RS-422/RS-232 serial communication cable.
- 25-pin connector to appropriate GPI/O connections.
- Video cable(s), up to 5: Coax/BNC for each of the available SDI inputs/outputs used.
- DB9 to BNC breakout cable for analog monitoring output(s). Available as an option from Chyron.
- Video monitor with RGB component or composite video input (optional). See §1.5 for more information on DigiBox CODI's analog video monitor outputs.

The cables are not included.

1.3 EAS DigiBox CODI Specifications and Diagrams

Video Inputs, SDI	• Video In • Key In
Video Outputs, SDI	• Video Out • Key Out
Monitor Output, analog	• RGB video, with separate sync connection and • Component video
Power Consumption	2A @ 120 V 1A @ 240 V (Auto-voltage selection) 50 or 60 Hz (Backup power supply optional)
External Connections	• Serial RS-232 or 422 (switchable): 38,400 baud max • 1 GPI/O connector, supporting up to 16 I/Os • Ethernet (CAT-5 10/100 Base-T); not implemented
Environmental	40° - 100° F (5° - 38° C) 5% - 95% Humidity (no condensation)
Dimensions	Height: 1.75” (4.45 cm) Width: 17” (43.2 cm) Depth: 11.5” (29.21 cm) Weight: 11 lbs, approximate

Figure 1-2 shows a block diagram of EAS DigiBox CODI's data and video connections.

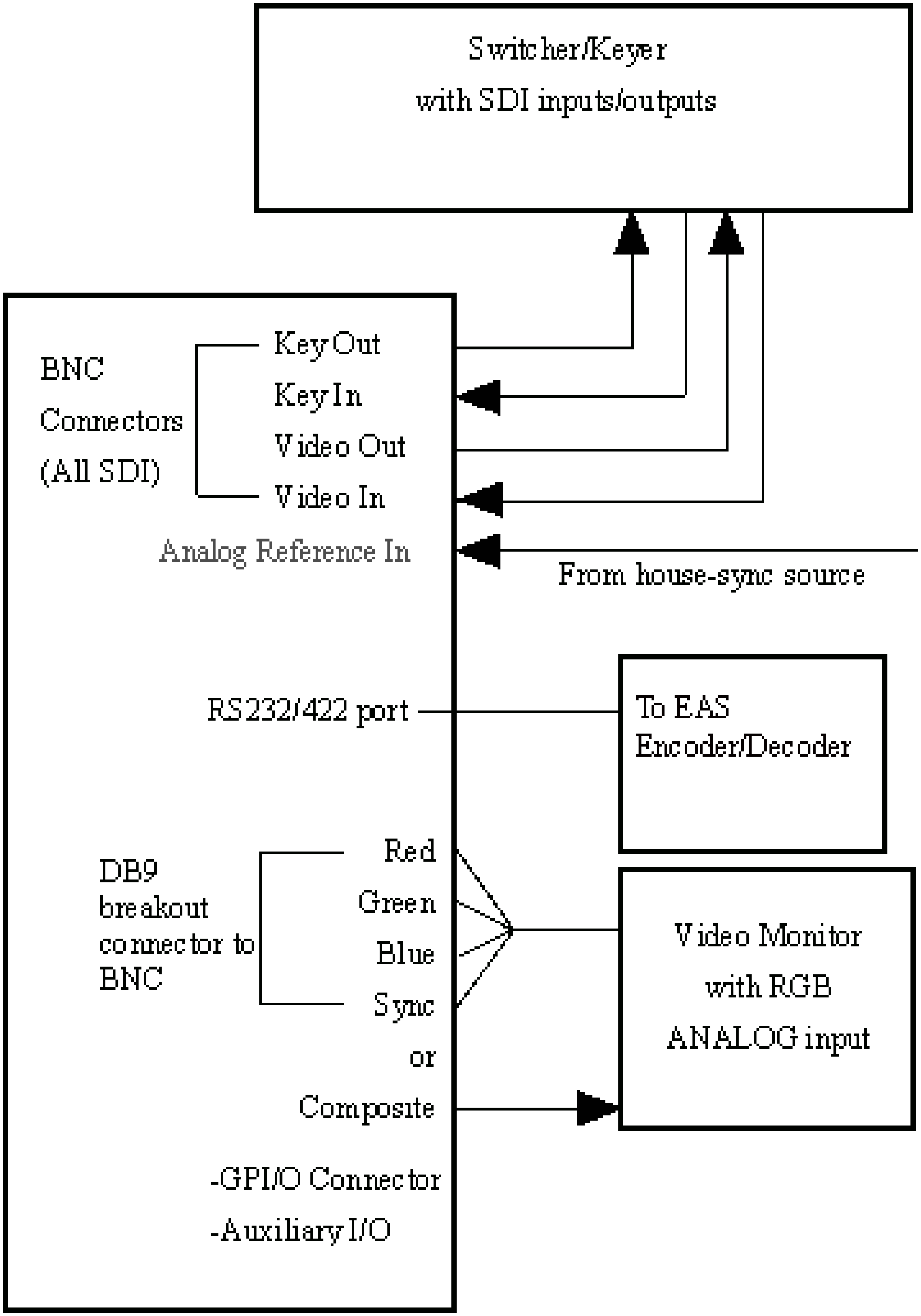
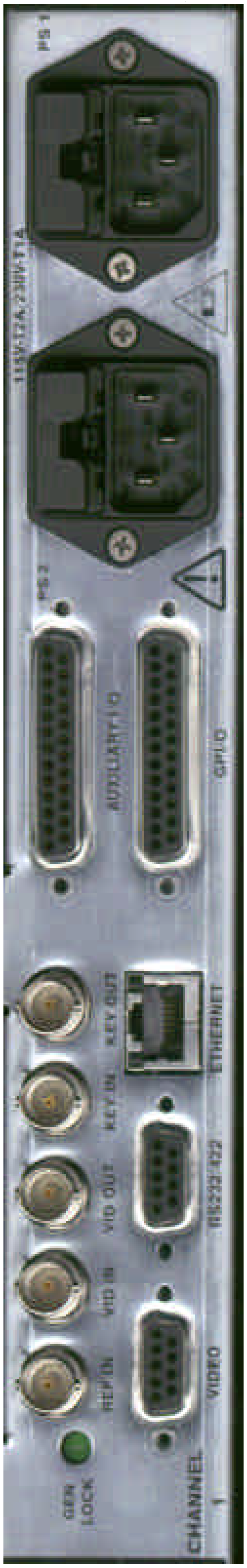


Figure 1-2. General EAS DigiBox CODI Diagram



All video, data and power connections are made from the back panel of the system, as seen here.

NOTES:

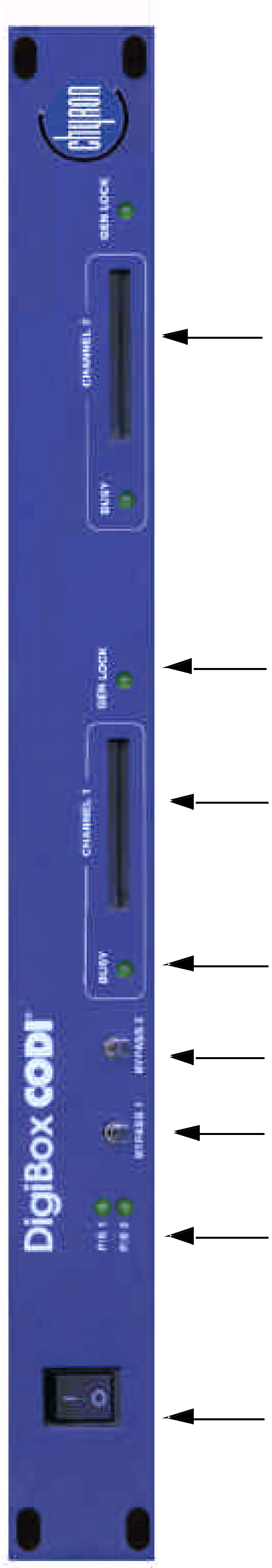
- The **AC Connector** closest to the Auxiliary I/O and GPI/O connectors feeds the ***optional*** back-up power supply. When a 2nd power supply is present in the system, both power supplies operate simultaneously, and one assumes full responsibility for supporting the system if the other power supply fails. The operation of each power supply is indicated by the LEDs marked “P/S 1” (and if present, “P/S 2”) on the front panel.

- The **25-pin Auxiliary I/O** connector is currently not used.

- The **Ethernet** connector is reserved for future use.

- A two-channel system or a second channel added to this chassis duplicates all of the connections (and the Gen Lock indicator light) but NOT the AC connectors.

Figure 1-3. Rear Panel, single-channel EAS DigiBox CODI



2nd Channel (optional)

GEN LOCK LED indicates that this channel of the system is receiving suitable sync input. Gives the same indication as the **GEN LOCK** LED on the back of the unit

Compact Flash disk slot (not used on EAS DigiBox CODI)

BUSY LED indicates that data is being retrieved from or written to the Compact Flash disk.

BYPASS switch for optional 2nd channel

BYPASS switch routes VIDEO IN directly to VIDEO OUT. (DOWN position = BYPASS mode)

PS1 and **PS2** LEDs show that each of the power supplies is functioning (second power supply optional).

Main power switch

Figure 1-4. EAS DigiBox CODI, shown with second-channel option

1.4 Data Connections

Note that EAS DigiBox CODI does not use the Ethernet connector that is present on the chassis. This document details DigiBox CODI's serial connections.

1.4.1 RS-232/RS-422 Selection

EAS DigiBox CODI's serial communication mode may be switched between RS-232 and RS-422 protocols with a slide switch located on the bottom of the chassis (pictured in Figure 1-5). NOTE THAT MANY OF THE LEADING EAS PRODUCTS USE THE RS-232 PROTOCOL.

BEFORE YOU DO ANYTHING ELSE, DETERMINE WHETHER YOU WILL BE USING AN RS-232 OR RS-422 SERIAL CONNECTION AND SET THIS SWITCH APPROPRIATELY! MIS-MATCHING RS-232 AND RS-422 CONNECTIONS MAY RESULT IN EQUIPMENT DAMAGE!

Always pay careful attention to the communications mode selected before connecting a serial cable to an external piece of equipment! (See Figure 1-6 for connector diagrams).

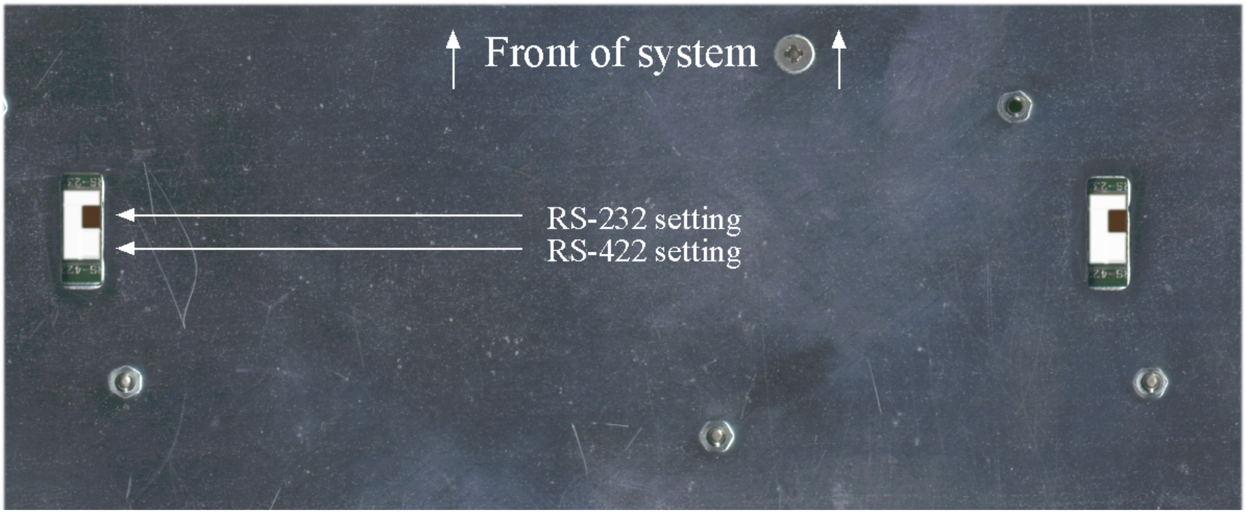


Figure 1-5. Serial communication protocol switches on bottom of chassis; system shown includes optional second channel (at right)

EAS DigiBox CODI Intelligent Interface communications are currently programmed to handle asynchronous word formats only (8 bits/no parity/1 stop).

1.4.2 Making or Repairing Serial Cables

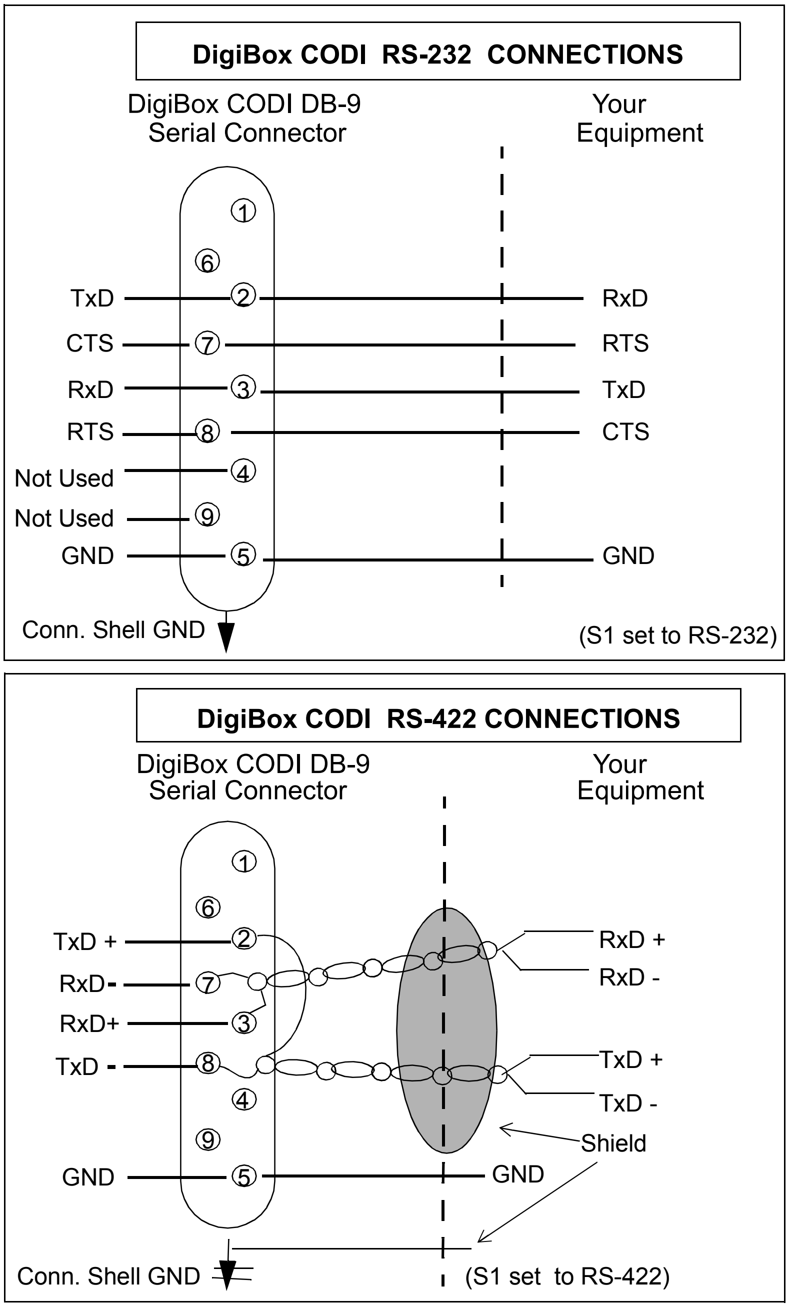


Figure 1-6. RS-232 and RS-422 Configurations

NOTE

When making your own cables please note that the three-wire interface, TX, RX & GND, does not always work. You may need to hard-wire the handshaking signals DSR and CTS to RTS to enable the proper transmission from the PC.

1.4.3 Serial Cabling To EAS Encoder/Decoder

The default RS-232 communication mode is described in the following sections. EAS DigiBox CODI's serial port uses a standard DCE DB-9 pinout. A 1:1 serial cable is provided to interface with a PC-AT -type serial port.

Figure 1-7 and Figure 1-8 illustrate connections to a 9-pin PC-AT type connector and to a 25-pin DTE type connector, respectively.

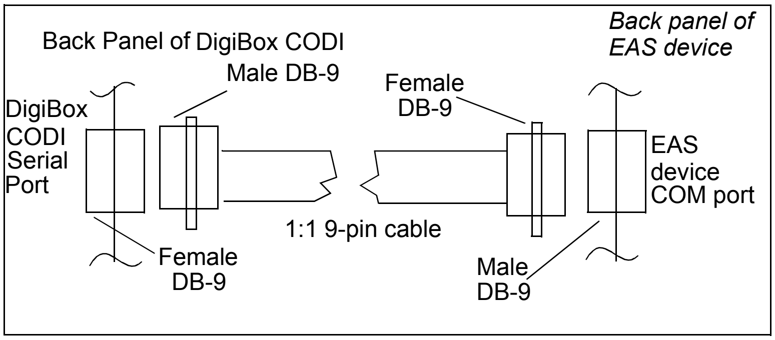


Figure 1-7. RS-232 Cabling with PC-AT-type connection

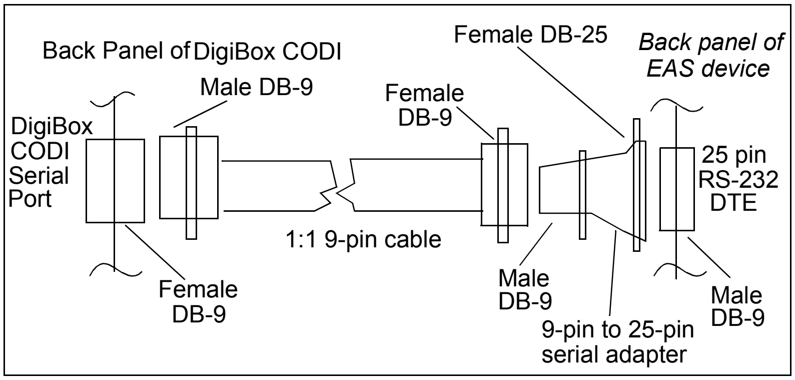


Figure 1-8. RS-232 Cabling with a standard 25-pin DTE interface

1.4.4 Serial Communication Settings

EAS DigiBox CODI is factory-configured to use the RS-232 protocol at 9600 baud. The baud rate may be changed in software, but extreme caution is recommended in undertaking this procedure. §2.1 in this document describes editing the appropriate file to make the setting. Other asyncronous Tx/Rx parameters such Data Bits, Stop Bits and Parity are fixed as listed below. The EAS receiver connected to EAS DigiBox CODI must be configured to match these parameters.

- 9600 baud
- 8 Data Bits
- 1 Stop Bit
- No Parity
- Half Duplex Operation (local display of characters).

The other communication options need not be set at this time, as they can run in default or off mode.

1.5 Video Monitor Hookup

Analog preview of DigiBox CODI's output is available at the 9-pin output connector on the back panel, shown in Figure 1-9.

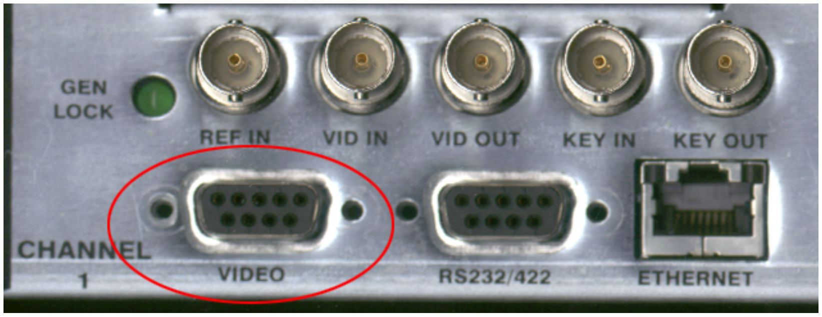


Figure 1-9. Detail, analog video connection for monitoring

NOTES

This connector supplies BOTH Component and Composite analog video signals.

This analog output provides a broadcast-quality signal. HOWEVER, take note that DigiBox CODI's internal keyer effect is ALWAYS present in this signal and CANNOT be turned off.

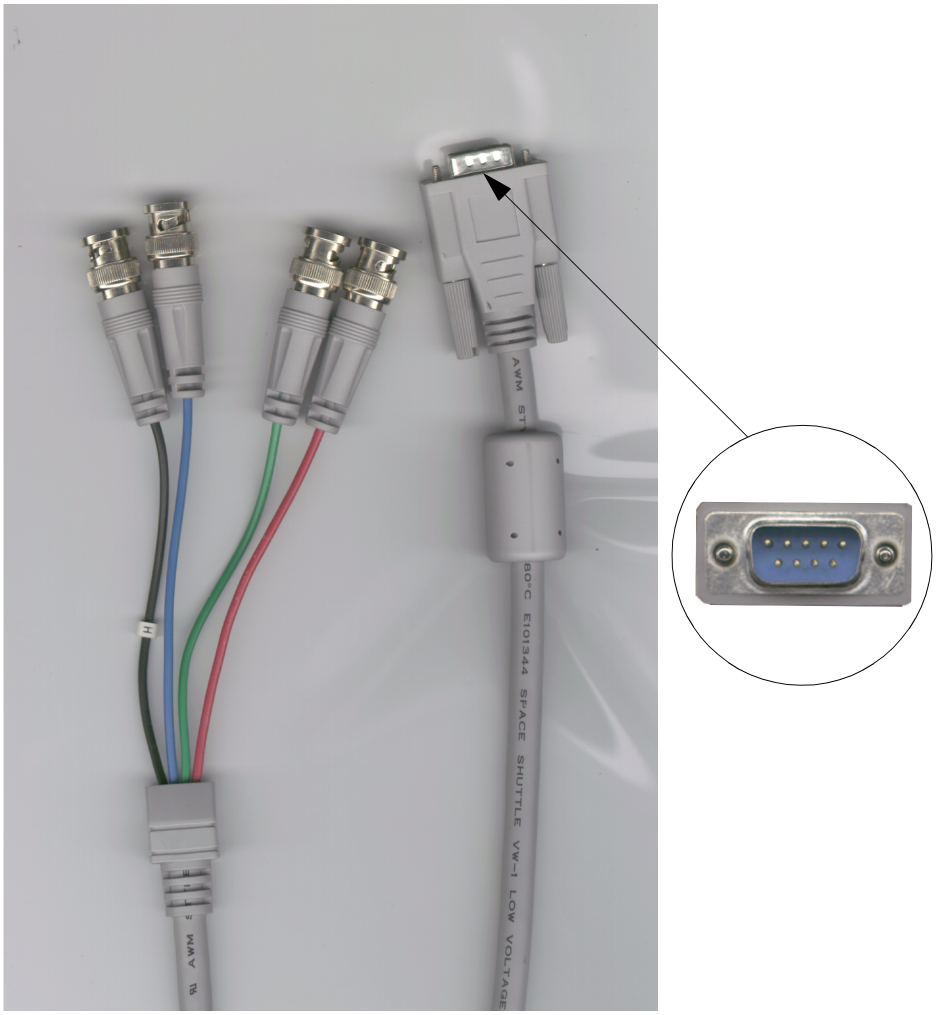


Figure 1-10. Optional cable for analog monitor output, front view of DB9 male connector at right

As noted above, both Composite and RGB Component video are output at all times from the nine-pin connector shown in Figure 1-9. Composite video is output by the black wire.

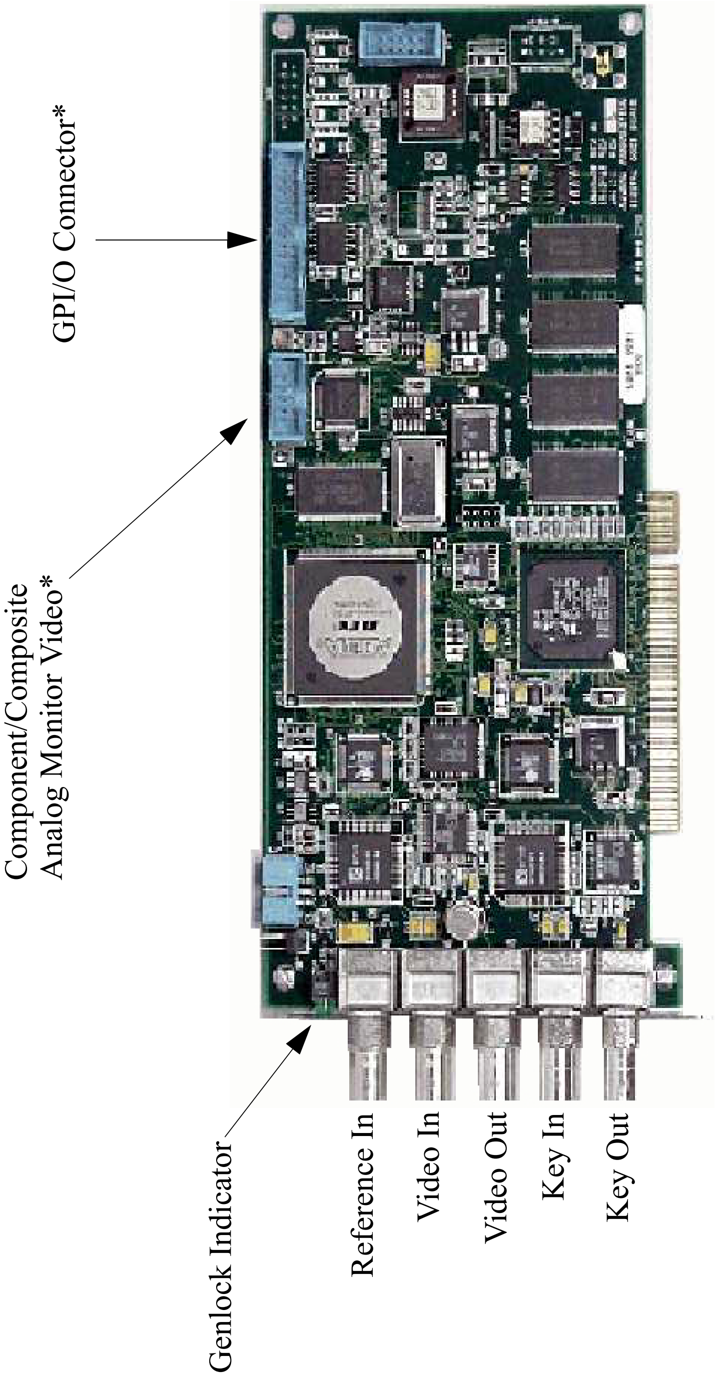
For RGB video, the red, green and blue wires output red, green and blue signals respectively. The signal at the black wire may be used as black burst for the composite signal.

CONNECTIONS ON REAR PANEL VIDEO CONNECTOR

1	RED SIGNAL	6	RED SHIELD
2	GREEN SIGNAL	7	GREEN SHIELD
3	BLUE SIGNAL	8	BLUE SHIELD
4	COMPOSITE VIDEO	9	COMPOSITE SHIELD
5	NO CONNECT		

1.6 Video Input/Outputs

The Video Input/Output connector configuration of the DigiBox CODI board rear mounting bracket is shown in Figure 1-11.



*Note that the Analog Monitor Video and GPI/O connectors shown here use ribbon cables to send their signals to a baseboard, which contains the external Video, RS232/422 and Ethernet connectors seen in Figure 1-8.

Figure 1-11. DigiBox CODI board I/O

All DigiBox CODI video inputs are fixed terminated inside the unit with standard 75-ohm termination. If input video must be routed to multiple destinations, a distribution amplifier should be used on the video source. All inputs are SDI 525/625-line compatible.

As shown in Figure 1-11, the DigiBox CODI has five Input/Output BNC connectors and an LED indicator on the rear panel. These are described in the table below:

INPUT/OUTPUT	FUNCTION
REF IN	This connector is used to connect an analog GENLOCK source input if digital GENLOCK is not used. The preferred input here is Black Burst.
VIDEO IN	Used for internally keying the DigiBox CODI video over incoming video. Can also be used as a digital GENLOCK source.
VIDEO OUT	Output of either keyed or non-keyed DigiBox CODI video.
KEY IN	Key source input from an external source or from another DigiBox CODI in multiple DigiBox CODI applications. In two-channel systems, this input may be taken from the other channel.
KEY OUT	Provides a key signal to external devices or the other DigiBox CODI board(s) in multiple DigiBox CODI applications. In two-channel systems, this output may feed the other channel.
The **GenLock** LED located above the REF IN connector indicates that the system (or the indicated channel of a two-channel system) is receiving a suitable sync signal.	

1.6.1 Selection of Genlock Source

DigiBox CODI can GENLOCK to either a digital or an analog source. Digital GENLOCK is taken from the signal connected to VIDEO IN. Analog GENLOCK signals should be connected to the REF IN connector. The **\VG** command must be issued to the board in order to select the proper reference. EAS DigiBox CODI systems ship from the factory set for **digital lock**. Any change made with the **\VG** command will be retained when the system is shut off.

1.6.2 Keying Over Live Video

Keying the DigiBox CODI graphics over live video can be accomplished in two ways; Internal Keying or External Keying.

- **Internal Keying**: This method of keying is also called *Downstream Mode.* External video is supplied to the DigiBox CODI and the CODI graphics are combined with the video stream and made available at the VIDEO OUT connector.
- **External Keying**: This method of keying is also called *Upstream Mode.* In this mode, Video signals and Key signals are sent to external equipment via the VIDEO OUT and KEY OUT connectors, respectively. Utilizing these signals, an external device, such as another DigiBox CODI, can mix (combine) the CODI graphics signal with the video stream.

1.6.3 Video Bypass Relay

Each DigiBox CODI channel contains a video bypass relay that controls the transfer of video if power is lost. In normal operation, the relay is engaged and the system functions as described previously. If power to the system is lost, the relay closes a set of contacts which allow full video to pass through the inactive system to the VIDEO OUT connector. When this occurs, a momentary video artifact will be seen. When power is returned and the system is subsequently re-booted, video bypass is de-activated and signal processing resumes.

1.7 GPI/O

DigiBox CODI uses up to sixteen General Purpose Interface (GPI) inputs for remote sensing of events or switch closures to trigger user-defined functions. The GPI connector on the rear panel of DigiBox CODI uses a standard DB-25 connector. All GPI/O configuration is accomplished via the serial port software.

Only GPI/O #1 is available in EAS DigiBox CODI; it has a pre-assigned function. See §2.2 for more information.

The correct pinouts for DigiBox CODI's GPI/O connector are shown below.

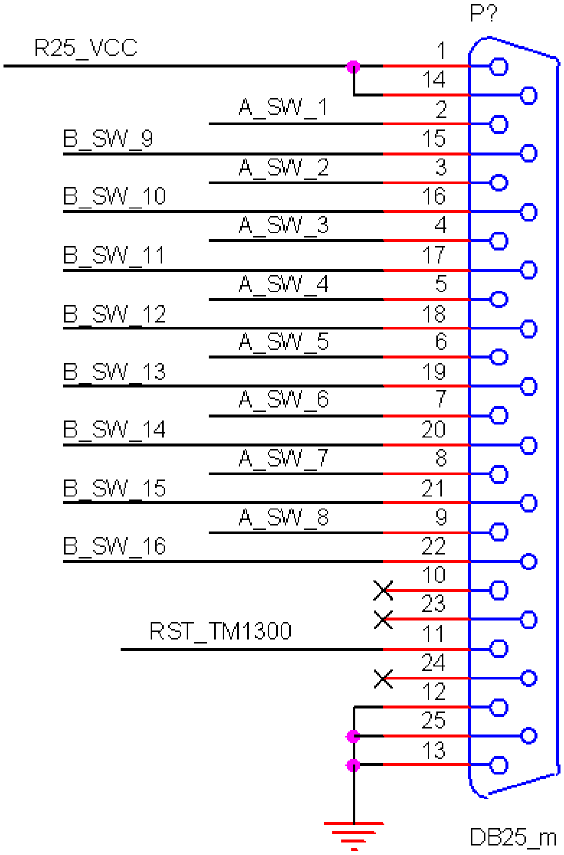


Figure 1-12. GPI/O connector pinout

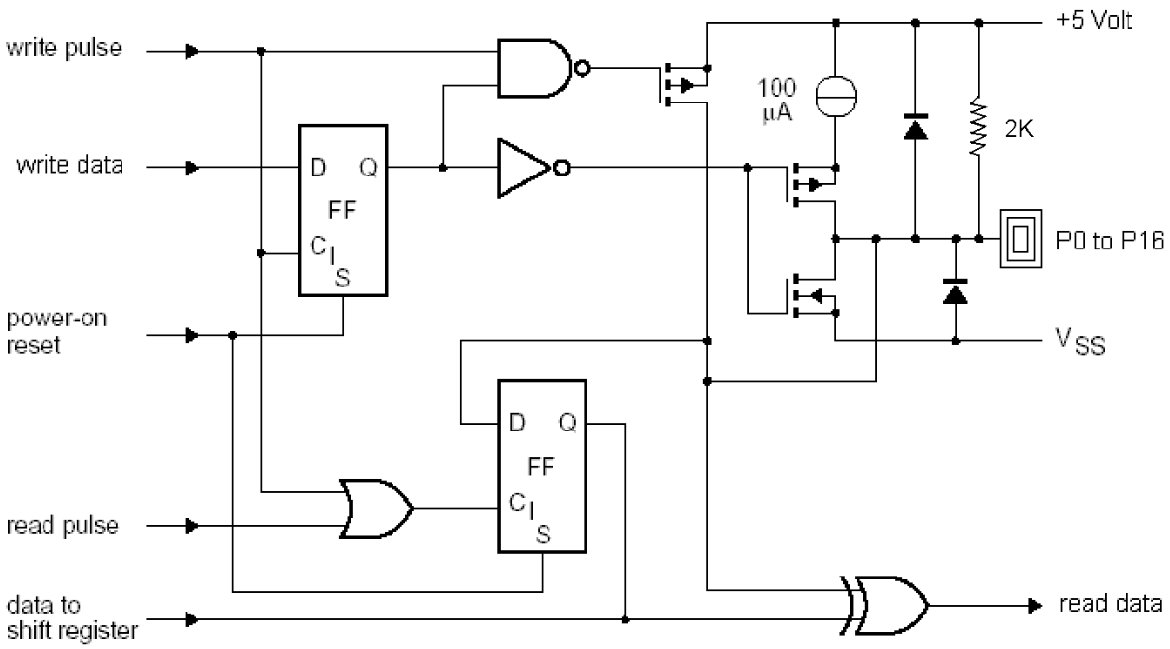


Figure 1-13. GPI/O Equivalent Circuit

Section 2: Initial Operation

2.1 Working With EAS DigiBox CODI's Software

The CompactFlash card slots on the front of EAS-configured DigiBox CODI units **are not used**. EAS DigiBox CODI operates ONLY from files on the unit's **internal** CompactFlash card. (Each channel is self-contained and has its own internal CompactFlash memory.)

Users are **not encouraged** to modify these files. However, instructions for doing so are included in the following sections for situations where changes to the files must be made. The DigiBox CODI system does not provide an interface for accessing the system's CompactFlash memory cards. The cards must be loaded into a CompactFlash card writer/ reader connected to a PC.

2.1.1 About DigiBox CODI's CompactFlash Cards

EAS DigiBox CODI is supplied with a CompactFlash card in the system's **internal** slot (see Figure 2-3). The memory card contains fonts and the system's basic operating files. See §2.1.4.2 . CompactFlash cards are widely available with various amounts of memory capacity, ranging from 16 to 512 MB.

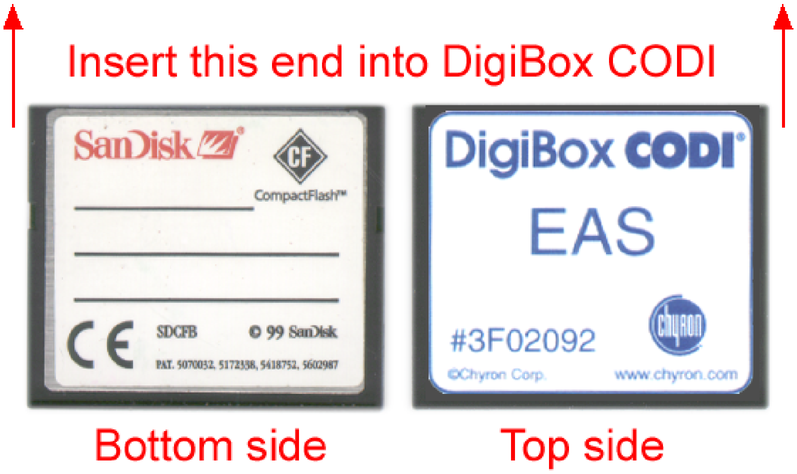


Figure 2-1. CompactFlash card, shown approximate size

2.1.2 Removing and Replacing the Internal CompactFlash Memory Card

Take note of the power supply which is to the right of the CompactFlash slots, as pictured below. Avoid contact with the power supply.

1. Turn the unit OFF, and disconnect the AC power cable.
2. Locate the internal CompactFlash memory card slot, as seen in Figure 2-2. Most of the slot itself is blocked from view by the Video Processor Board, which is mounted above the baseboard. However, the end of the card is accessible.

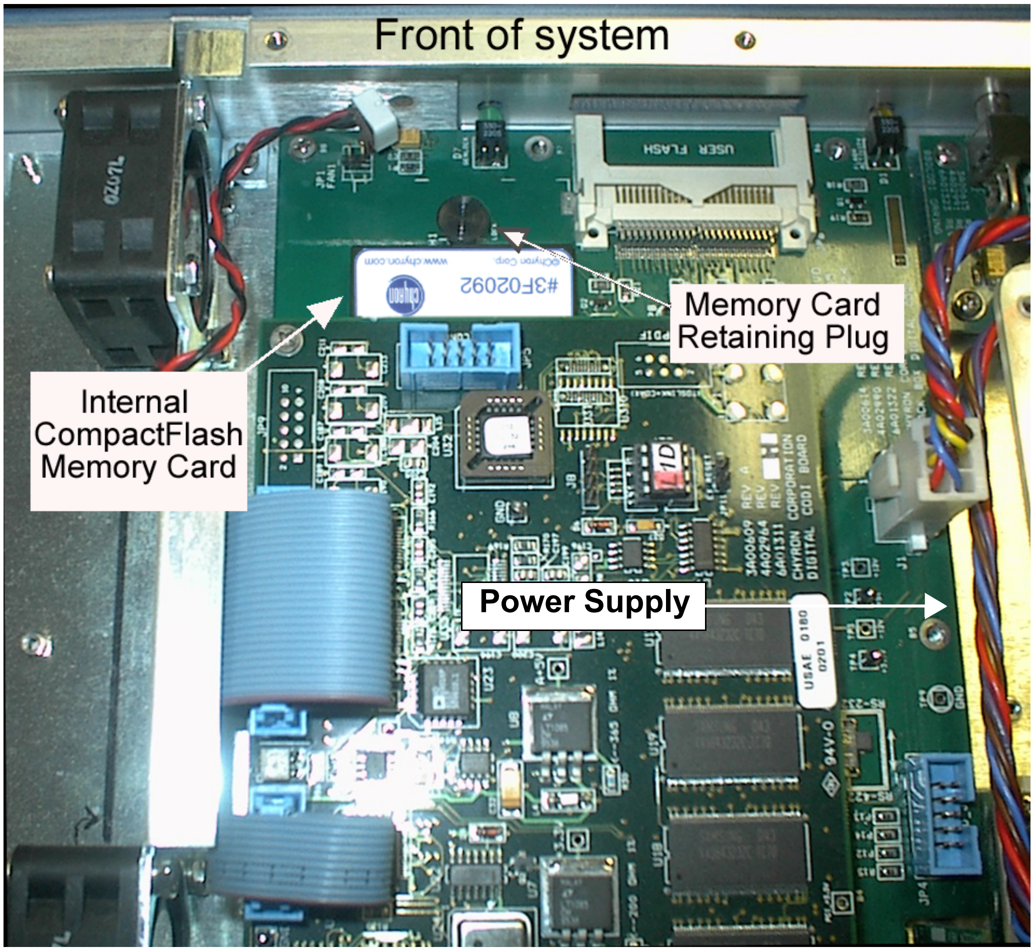


Figure 2-2. Internal CompactFlash Memory Card and Retaining Plug

3. Remove the memory card Retaining Plug by grasping it between thumb and forefinger and pulling it straight up. Don’t lose it.
4. Pull the CompactFlash memory card straight out of its slot.

Proceed to §2.1.3.

2.1.3 Working With CompactFlash Memory Cards

2.1.3.1 Writing To/Reading From CompactFlash Cards

Remember that DigiBox CODI can **only read** from CompactFlash cards. To edit the files contained therein, you must use a CompactFlash card reader/writer such as those pictured in Figure 2-3.

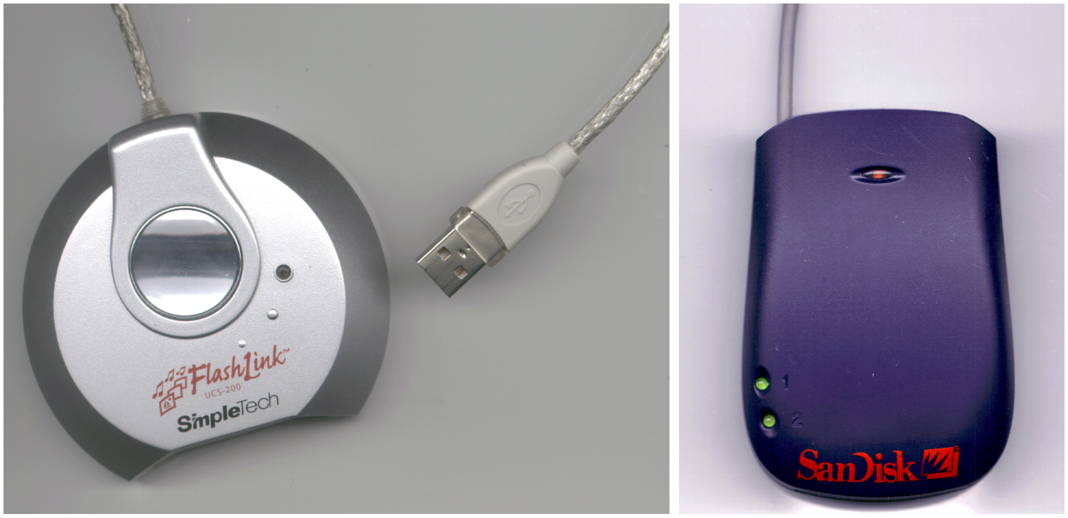


Figure 2-3. CompactFlash card writer/readers for connection to a PC via USB port

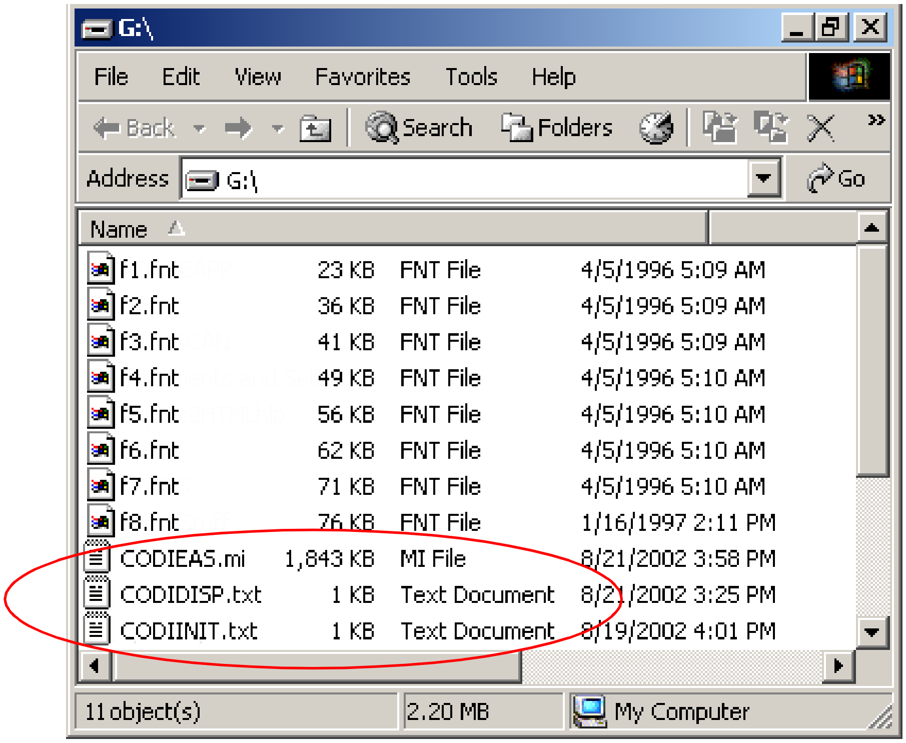
CompactFlash card writer/readers of this type are purchased separately, and are widely available from retailers of PC peripherals.

2.1.3.2 Formatting CompactFlash Cards For Use With DigiBox CODI

If you wish to make backups of your system's CompactFlash memory, take special note that CompactFlash cards for use with DigiBox CODI must be formatted with the FAT file system (NOT FAT32 or NTFS).

2.1.4 Editing EAS DigiBox CODI's Files

The internal CompactFlash memory card supplied with each EAS DigiBox CODI channel contains three important files, seen circled in Figure 2-4, along with system fonts. Note that the view shown is for informational purposes only. The contents of the Compact Flash card are seen here through a CompactFlash card reader connected via Windows to a PC.



Name			
f1.fnt	23 KB	FNT File	4/5/1996 5:09 AM
f2.fnt	36 KB	FNT File	4/5/1996 5:09 AM
f3.fnt	41 KB	FNT File	4/5/1996 5:09 AM
f4.fnt	49 KB	FNT File	4/5/1996 5:10 AM
f5.fnt	56 KB	FNT File	4/5/1996 5:10 AM
f6.fnt	62 KB	FNT File	4/5/1996 5:10 AM
f7.fnt	71 KB	FNT File	4/5/1996 5:10 AM
f8.fnt	76 KB	FNT File	1/16/1997 2:11 PM
CODIEAS.mi	1,843 KB	MI File	8/21/2002 3:58 PM
CODIDISP.txt	1 KB	Text Document	8/21/2002 3:25 PM
CODIINIT.txt	1 KB	Text Document	8/19/2002 4:01 PM

Figure 2-4. Files on the supplied Compact Flash card

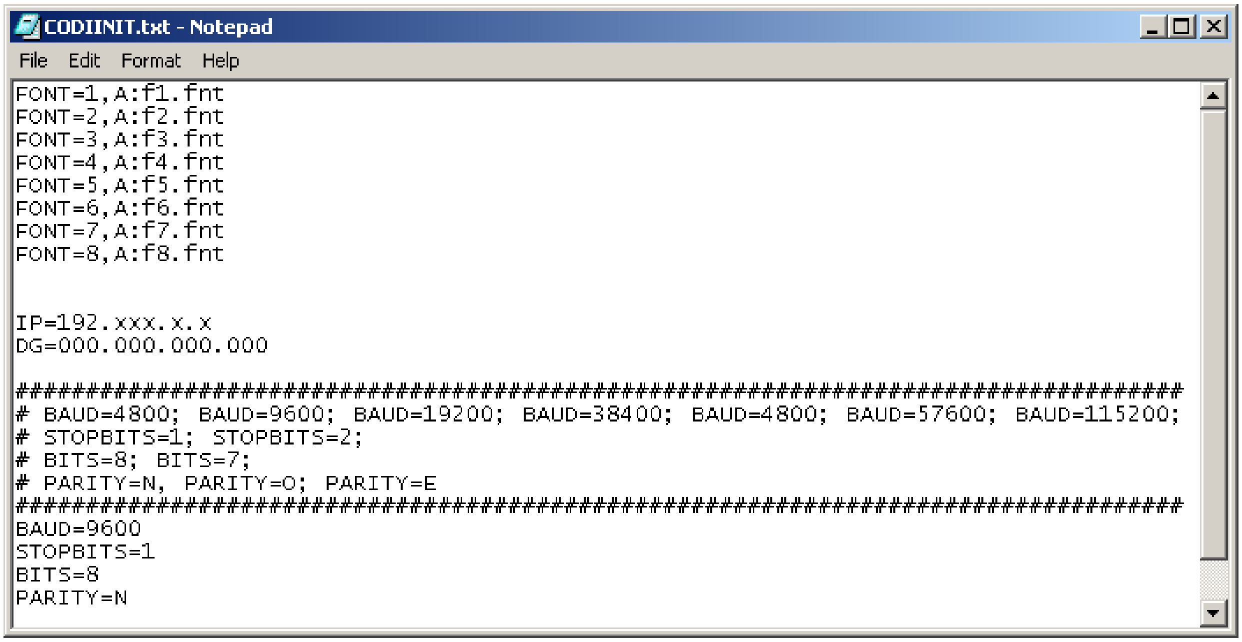
2.1.4.1 CODIEAS.mi And CODIDISP.txt

- **CODIEAS.mi** is the system's software and must not be edited or moved from the filepath in CompactFlash memory show in Figure 2-4.
- **CODIDISP.txt** may be edited by the user to automatically launch display tasks upon system startup. Use a program such as Windows Notepad to enter any of the commands found in Section 3 of this document.

2.1.4.2 Setting Serial Communication Speed by Editing CODIINIT.txt

As stated in §1.4.4, EAS DigiBox CODI is factory-set for a baud rate of 9600. This setting may be modified by editing the CODIINIT.TXT file.

1. Locate the file and open it in an application such as Notepad, as seen below.



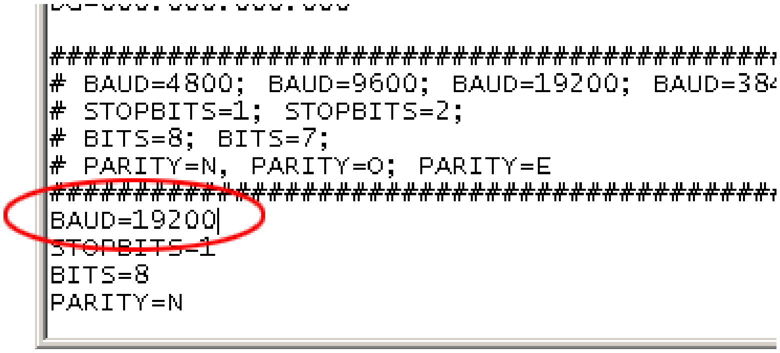
```
FONT=1,A:f1.fnt
FONT=2,A:f2.fnt
FONT=3,A:f3.fnt
FONT=4,A:f4.fnt
FONT=5,A:f5.fnt
FONT=6,A:f6.fnt
FONT=7,A:f7.fnt
FONT=8,A:f8.fnt


IP=192.xxx.x.x
DG=000.000.000.000

####################################################################################
# BAUD=4800; BAUD=9600; BAUD=19200; BAUD=38400; BAUD=4800; BAUD=57600; BAUD=115200;
# STOPBITS=1; STOPBITS=2;
# BITS=8; BITS=7;
# PARITY=N, PARITY=O; PARITY=E
####################################################################################
BAUD=9600
STOPBITS=1
BITS=8
PARITY=N
```

Figure 2-5. CODIINIT.txt file, ready for editing

2. To change the baud rate, enter new values directly into the file's text, as shown below:



```
####################################################
# BAUD=4800; BAUD=9600; BAUD=19200; BAUD=38
# STOPBITS=1; STOPBITS=2;
# BITS=8; BITS=7;
# PARITY=N, PARITY=O; PARITY=E
####################################################
BAUD=19200
STOPBITS=1
BITS=8
PARITY=N
```

Figure 2-6. Changing BAUD rate by editing the file CODIINIT.txt

3. Save the file back to its original location on the CompactFlash memory card.
4. Remove the card from the CompactFlash Writer/Reader and return it to its slot in the EAS DigiBox CODI.
5. Replace the memory card Retaining Plug.
6. Restore power and data connections to the unit.
7. Turn the unit on and confirm that it is operating correctly.

2.2 Operation

EAS DigiBox CODI can serve as a direct replacement for Chyron's "Black Box" CODI systems in Emergency Alert System installations. EAS DigiBox CODI's serial port must be connected to an encoder/decoder device as mentioned in §1.2 and §1.4.1. EAS encoder/decoders are constantly connected to external networks to receive EAS notifications at any time. These devices also contain CODI commands in firmware; when the encoder/decoder receives an EAS notification, it triggers DigiBox CODI to begin displaying the incoming EAS information as a crawl. Activation of the display is typically accompanied by a pre-set automatic keyer effect that adds the display at the bottom of the station's transmitted picture.

EAS-configured DigiBox CODI installations use **GPI #1** for a "commercial killer" feature written into the system's software. Contrary to its unofficial name, this feature is actually used to suspend the keyed-in EAS display during commercial breaks.

This GPI/O's normal state is "open", placing EAS DigiBox CODI's keyer **in** the program signal path during all feature programming. In this manner, an EAS display is seen immediately when the keyer effect is activated. During commercial breaks, GPI #1 is closed by master control automation (or the master control operator), thereby insuring that these breaks air without the EAS display.

Consult the documentation that accompanies your facility's EAS encoder/decoder for more information.

Section 3: DigiBox CODI Commands

3.1 INTRODUCTION

This section provides describes the command structure and operations of the DigiBox CODI. Each command entry provides the command format, an example of how to apply the command and a brief functional description. The different types of commands covered in this section are listed below:

- **Background Commands** (§ 3.3, on 3-5)
- **Color Commands** (§ 3.4, on 3-17)
- **Display Commands** (§ 3.5, on 3-23)
- **Event Timer Commands** (§ 3.6, on 3-29)
- **Flash Commands** (§ 3.8, on 3-45)
- **Font Commands** (§ 3.9, on 3-50)
- **GPIO Commands** (§ 3.10, on 3-56)
- **Graphics Commands** (§ 3.11, on 3-64)
- **Letter Commands** (§ 3.12, on 3-71)
- **Line Draw Commands** (§ 3.13, on 3-78)
- **Message Commands** (§ 3.14, on 3-84)
- **Message Effect Commands** (§ 3.15, on 3-94)
- **Palette Commands** (§ 3.16, on 3-114)
- **Page Commands** (§ 3.17, on 3-119)
- **Row Commands** (§ 3.18, on 3-127)
- **System Commands** (§ 3.19, on 3-140)
- **Tab Commands** (§ 3.20, on 3-160)
- **Video Commands** (§ 3.21, on 3-169)
- **Watch Commands** (§ 3.22, on 3-186)
- **Miscellaneous Commands** (§ 3.23, on 3-195)

If you are currently using the analog “Black Box”CODI character generator, you will find that the DigiBox CODI uses the same command sets as the analog version. Programs written for the Black Box CODI can easily be recompiled using the DigiBox CODI library and DLL modules. In effect, this commonality makes for a more straightforward conversion process to the DigiBox CODI and for quicker start-up time.

3.2 DigiBox CODI Command Features

The DigiBox CODI provides the following additional command features:

- Key In/Out, Video In/Out, Genlock, Analog Monitor Out, eight (16) external GPIO triggers and a video bypass relay.
- Each pixel is in 32-bits, RGBA format
- Separate Foreground and Background Buffers
- Can read directly from and write directly to Windows files. Many commands, like **Download Font** and **Download Graphic**, can contain a filename from which the pcCODI can directly retrieve data. It is no longer necessary to pass data via the pcCODI DLL command.
- Provides separate background memory to store full-color images.
- Automatically anti-aliases with both Background and Video.
- Can be chained with the Video and Key Out from one board to plug into the Video and Key In of another board.
- Can read *iNFiNiT!* fonts, graphics, and messages. Original content created on an *iNFiNiT!, MAX!* or *MAXINE!* can be displayed on the DigiBox CODI. *iNFiNiT*! tab fields are automatically converted to pcCODI tab fields in order to update their content.
- Includes a built-in Boot Loader in the DLL to load DigiBox CODI embedded code from your application.
- Includes a built-in Font Creation command to create a pcCODI font from a TrueType font format.
- Uses 16 external GPIO triggers to call functions from your application or provide an external trigger for other devices.
- Can load up to 16 on-line fonts.
- Uses a Windows console-style diagnostic screen.
- Provides 256 levels of anti-aliasing using *iNFiNiT!* fonts.
- Uses the fast PCI Bus of the PC.
- Uses a 32-bit Windows driver
- Has the ability, using the **Message Save** command, to save DigiBox CODI message(s) to a file on the hard drive. Since it does not save it in memory, this feature enables you to save an unlimited number of messages. The number of messages that can be saved is limited only by the size of the hard drive.

3.2.1 Commands Not Used

Commands that are not used by the DigiBox CODI include those employed in video setup on analog CODI systems and the color mapping required in the 8-bit/pixel format. Since the DigiBox CODI Video Processor Board is 601-compliant, and each pixel is in a 32-bit format, these commands are no longer relevant. It is not necessary that you remove these commands from your program code, but take note that they will be ignored by DigiBox CODI. The existing commands that are not supported by DigiBox CODI are listed in Table 3-1.

Table 3-1: Commands not Supported by DigiBox CODI

Command Name	Format
Background Setup (Interactive)	`\BS\\`
Color Background	`\CB\color[\color…]\\`
Color Video	`\CV\color[\color…]\\`
Upload Color	`\UC\index\\`
Message Blank	`\MB\message\\`
Message Effect Zoom	`\ME\ZO\message\direction\speed\steps\\`
Video Horizontal Phase	`\VH\[+,-]hphase\\`
Video Key Delay	`\VK\[+-]delay\\`
Video Subcarrier Phase	`\VS\[+-]sphase\\`

3.2.2 Command Notation

Command arguments shown in UPPERCASE letters are required fields and must be entered in the same format as they appear in this manual. Command arguments shown in lower case letters must be replaced by one of the values specified. Arguments stated within brackets [ ] are mandatory. Arguments stated within brackets { } are optional. Arguments stated within brackets { } and followed by ellipsis... can be repeated as many times as necessary.

3.3 Background Commands

The DigiBox CODI background is arranged in scanline pairs. Each scanline pair can be keyed for either a DigiBox CODI-generated background or an external video display. When a background scanline pair is keyed for a DigiBox CODI-generated background, a DigiBox CODI background color is displayed from a palette of 16.7 million colors.

Background commands operate on a separate Background Frame Buffer in the DigiBox CODI board. Because of this, there is a slight delay when Background functions are used because of the time it takes to load the Background memory. This is normal and does not affect operation.

The background commands that are described in this paragraph are:

• **Download Background (HSI)**	**(§ 3.3.1, on 3-6)**
• **Download Background (RGB) Mode 1**	**(§ 3.3.2, on 3-7)**
• **Download Background (RGB) Mode 2**	**(§ 3.3.3, on 3-8)**
• **Download Background (HSI) Mode 2**	**(§ 3.3.4, on 3-9)**
• **Download Background (File-Blend) Mode 3**	**(§ 3.3.5, on 3-10)**
• **Download Background (File-Cut) Mode 4**	**(§ 3.3.6, on 3-11)**
• **Download Background (PCI-Blend) Modes 5,6,7**	**(§ 3.3.7, on 3-12)**
• **Download Background (PCI-Cut) Modes 8 - 11**	**(§ 3.3.8, on 3-13)**
• **Upload Background (HSI) Mode 2**	**(§ 3.3.9, on 3-14)**
• **Background Erase**	**(§ 3.3.10, on 3-15)**
• **Set Scanline of Background Buffer**	**(§ 3.3.11, on 3-16)**

An NTSC DigiBox CODI contains 243 displayed scanline pairs (486 scanlines in total). The first 243 colors of the background palette are displayed for the 243 scanline pairs. A PAL DigiBox CODI contains 288 displayed scanline pairs (576 scanlines total). The Download Background commands define the background keyer and palette entry values. These commands allow you to define each background scanline pair color and key individually, or to create a background ramp from a start to an end scanline pair using a start and end color.

The Upload Background command allows you to upload the current DigiBox CODI background palette and keyer to the host. The upload data can be saved on the host for future use.

3.3.1 Download Background (HSI) Mode 1, 7

Format: `\DB\mode\H\h\s\i\k{\h\s\i\k...}\\`

Parameters:

mode = 1 (background loading)
`h` = hue (0 - 360)
`s` = saturation (0 - 255)
`i` = intensity (0 - 255)
<u>mode 1:</u>
`k` = key (0, 1)
0 = display background
1 = display external video
<u>mode 7:</u>
`k` = key (0 - 255)

`\h\s\i\k` parameters repeated: NTSC=256, PAL=271

Return: none

Description:

The Download Background (HSI) command, in mode 1, is used to specify the color values for each scanline pair. In the HSI format, each scanline pair is defined by a hue, saturation, intensity value and a key. An NTSC DigiBox CODI has 256 scanline pairs, while a PAL DigiBox CODI has 271 pairs. Therefore, the background download information must consist of 256 NTSC sets of four values (271 sets for PAL).

3.3.2 Download Background (RGB) Mode 1, 7

Format: `\DB\mode\R\r\g\b\k{\r\g\b\k...}\\`

Parameters:
mode1 (background loading)
`r` = red (0 - 255)
`g` = green (0 - 255)
`b` = blue (0 - 255)
<u>mode 1:</u>
`k` = key (0, 1)
0 = display background
1 = display external video
<u>mode 7:</u>
`k` =key (0 - 255)

`\r\g\b\k` parameters repeated NTSC=256, PAL=271

Return: none

Description

The Download Background (RGB) command, in mode 1, is used to specify the color values for each scanline pair. In the RGB format, each scanline pair is defined by a red, green, blue value and a key. An NTSC DigiBox CODI has 256 scanline pairs, while a PAL DigiBox CODI has 271. Therefore, the background download information must consists of 256 NTSC sets of four values (271 sets for PAL).

3.3.3 Download Background (RGB) Mode 2, 8

Format: `\DB\mode\R\start\r\g\b\k\end\r\g\b\k\\`

Parameters:

`mode` = 2 (background create)
`start` = start scanline for generation
(NTSC: 1 - 256, PAL: 1 - 271)
`end` =end scanline for generation
(NTSC: 1 - 256, PAL: 1 - 271)
`r` = red (0 - 255)
`g` = green (0 - 255)
`b` = blue (0 - 255)

mode 2:
- **`k`** = key (0, 1)
  - 0 = display background
  - 1 = display external video

mode 8:
- **`k`** =key (0 - 255)

Return: none

Description:

The RGB Download Background command, in mode 2, is used to create backgrounds. With this command, it is possible to ramp from one color to another automatically. In order to do this, you must specify the starting scanline pair and color, and the ending scanline pair and color. The scanlines between the start and end lines are colored with a ramp between the start and end colors. In the RGB format, you must specify the start and end red, green, blue values and key.

3.3.4 Download Background (HSI) Mode 2, 8

Format: `\DB\mode\H\start\h\s\i\k\end\h\s\i\k\\`

Parameters:
`mode` = 2 (background create)
`start` = start scanline for generation
(NTSC: 1 - 256, PAL: 1 - 271)
`end` = end scanline for generation
NTSC 1 - 256, PAL 1-271)
`h` = hue (0 - 360)
`s` = saturation (0 - 255)
`i` = intensity (0 - 255)
mode 2:
`k` = key (0, 1)
0 = display background
1 = display external video
mode 8:
`k` =key (0 - 255)

Return: none

Description:

The HSI Download Background command, in mode 2, is used to create backgrounds. Using this command, it is possible to ramp from one color to another automatically. In order to do this, you must specify the starting scanline pair and color, and the ending scanline pair and color. The scanlines between the start and end lines are colored with a ramp between the start and end colors. In the HSI format, you must specify the start and end hue, saturation, intensity values and key. You can define multiple ramps for the background in a single command.

3.3.5 Download Background (File-Blend) - Mode 3

Format: `\DB\mode\X position\Y position\`
`[filename]\height\width\\`

Parameters: `mode` = 3

`X position, Y position` = the coordinates in which the Graphic should be placed in the background:

X position range (0 - 719)
Y position range (NTSC: 0 - 485, PAL: 0 - 575)

NOTE

For a graphic that covers the entire background, these values should be set as follows: `X position =` 0,0, while `Y position` values cannot be negative:

`[filename]` = the path and name of the file in which the graphic can be found. The type of file to be loaded will be determined by the filename extension. The following is a list of valid extensions:

Extension		Graphic File Format
.chy	=	Chyron RGBA
.TIF	=	Tif file
.TGA	=	TARGA
.bmp	=	Bitmap file

`height and width` = optional parameters that provide Automatic Scaling of the graphic before it is displayed. When setting height and width values, DigiBox CODI will scale the graphic to that dimension. If you want the scaling to be proportional, assign a value to one of the parameters, then set the other parameter to 0. For example, if you want to scale an incoming graphic to 50 scanlines, set the height to 50 and the width to 0. In this way, the original proportions of the graphic will be maintained. If the height and width parameters are omitted, the original size of the graphic will be used.

Return: File not Found
Invalid Parameters

Description:

The Download Background (File-Blend) command allows you to blend a full color graphic onto a background. By reading in the graphic file from disk, DigiBox CODI then uses the graphic's Alpha values to blend the graphic accordingly. Using these parameters, you are able to layer multiple images onto a single background.

3.3.6 Download Background (File-CUT) - Mode 4

Format: `\DB\mode\X position\Y position\`
`[filename]\height\width\\`

Parameters: `mode` = 4

`X position, Y position` = the coordinates in which the Graphic should be placed in the background. X position range = 0 - 719; Y position range = NTSC: 0 - 485, PAL: 0 - 575)

NOTE

For a graphic that covers the entire background, these values should be set as follows: `X position =` 0,0, while `Y position` values cannot be negative:

`[filename]` = the path and name of the file in which the graphic can be found. The type of file to be loaded will be determined by the filename extension. The following is a list of valid extensions:

Extension		Graphic File Format
.chy	=	Chyron RGBA
.TIF	=	Tif file
.TGA	=	TARGA
.bmp	=	Bitmap file

`height and width` = optional parameters that provide Automatic Scaling of the graphic before it is displayed. When setting height and width values, DigiBox CODI will scale the graphic to that dimension. If you want the scaling to be proportional, assign a value to one of the parameters, then set the other parameter to 0. For example, if you want to scale an incoming graphic to 50 scanlines, set the height to 50 and the width to 0. In this way, the original proportions of the graphic will be maintained. If the height and width parameters are omitted, the original size of the graphic will be used.

Return: File not Found
Invalid Parameters

Description:

The Download Background (File-CUT) command allows you to cut a full color graphic into an existing background. By reading in the graphic file from disk, DigiBox CODI then uses the graphic's Alpha values to cut the graphic accordingly.

3.3.7 Download Background (PCI-Blend) - Modes 5, 6, and 7

Format: `\DB\mode\X position\Y position\height\width\\data`

Parameters: `mode` = **5** (read Chyron RGBA from PCI and blend with background.)
`mode` = **6** (read TIF from PCI and blend with background.)
`mode` = **7** (read TGA from PCI and blend with background.)

`X position, Y position` = the coordinates in which the Graphic should be placed in the background.

X position range (0-719)
Y position range (NTSC: 0-485, PAL: 0-575)

NOTE

For a graphic that covers the entire background, these values should be set as follows: `X position =` 0,0, while `Y position` values cannot be negative:

`height and width` = optional parameters that provide Automatic Scaling of the Graphic before it is displayed. When setting height and width values, DigiBox CODI will scale the graphic to that dimension. If you want the scaling to be proportional, assign a value to one of the parameters, then set the other parameter to 0. For example, if you want to scale an incoming graphic to 50 scanlines, set the height to 50 and the width to 0. In this way, the original proportions of the graphic will be maintained. If the height and width parameters are omitted, the original size of the graphic will be used.

`data` = the data from the graphic file that is read by the user's application and then sent to the DigiBox CODI via the ***writemempccodi( )*** function.

Return: Invalid Parameters

Description:

The Download Background (PCI Blend) command allows you to blend a full color graphic into an existing background. By reading in the graphic file from the PCI BUS, DigiBox CODI then uses the graphic's Alpha values to blend the graphic accordingly. Using these parameters, you are able to layer multiple images onto a single background.

NOTE

Data must be transmitted using the WriteMemPCCODI( ) function. DigiBox CODI will wait until all data files are received.

3.3.8 Download Background (PCI-CUT) - Modes 8, 9, 10, 11

Format: `\DB\mode\X position\Y position`
`\height\width\\data`

Parameters: `mode` = **8** (read Chyron RGBA from PCI and cut with background.)
`mode` = **9** (read TIF from PCI and cut with background.)
`mode` = **10** (read TGA from PCI and blend with background.)
`mode` = **11** (read BMP from PCI and blend with background.)

`X position, Y position` = the coordinates in which the Graphic should be placed in the background.

X position range (0-719)
Y position range (NTSC: 0-485, PAL: 0-575)

NOTE

For a graphic that covers the entire background, these values should be set as follows: `X position =` 0,0, while `Y position` values cannot be negative:

`height and width` = optional parameters that provide Automatic Scaling of the graphic before it is displayed. When setting height and width values, DigiBox CODI will scale the graphic to that dimension. If you want the scaling to be proportional, assign a value to one of the parameters, then set the other parameter to 0. For example, if you want to scale an incoming graphic to 50 scanlines, set the height to 50 and the width to 0. In this way, the original proportions of the graphic will be maintained. If the height and width parameters are omitted, the original size of the graphic will be used.

`data` = the data from the graphic file that is read by the user's application and then sent to the DigiBox CODI via the ***writemempccodi( )*** function.

Return: File not Found
Invalid Parameters

Description:

The Download Background (PCI Cut) command allows you to cut a full color graphic into an existing Background. By reading in the graphic file from the PCI BUS, DigiBox CODI then uses the graphic's Alpha values to cut the graphic accordingly. Using these parameters, you are able to layer multiple images onto a single background.

NOTE

Data must be transmitted using the WriteMemPCCODI( ) function. DigiBox CODI will wait until all data files are received.

3.3.9 Upload Background (HSI) Mode 2

Format: `\UB\mode\\`

Parameters: `mode` = **1** (upload mode):
1 = ASCII upload

Return: `\DB\7\R\r\g\b\k\...\r\g\b\k\\)`
`r` = red (0 - 255)
`g` = green (0 - 255)
`b` = (blue) (0 - 255)
`k` = key (0 - 255):

`\r\g\b\k\` parameters repeated NTSC: = 256,
PAL: = 271

Description:

The Upload Background command sends the palette, RGB, and key information for the current background to the host. The background is defined in scanline pairs. Each scanline pair is defined by a red, green, blue and key component. Therefore, the background upload information consists of 256 NTSC sets of four values, 271 sets for PAL.

3.3.10 Background Erase

Format: `\BE{\N}\\`

Parameters: `N` = Erase - non display (if this entry is omitted, the display will be erased.

Return: none

Description:

This function will erase the background buffer.

3.3.11 Set Scanline of Background Buffer

Format: `\BL\n\\`

Parameters: **`n`** = Raster scanline (0 - 63)

Return: none

Description:

This function sets the starting raster scanline for the background buffer. This is a hardware setting of where the starting scanline should be located for the background plane.

Normally, the default DigiBox CODI setting should suffice for most installations. This command should only be required if it is necessary to adjust for major video timing differences.

3.4 Color Commands

The DigiBox CODI maintains an internal palette of eight (8) colors for characters (text) and eight (8) colors for edges. These colors can contain any RGB value up to 16.7 million.

All graphics, backgrounds and logos can be full color and are separate from the 8 character colors

The commands that are described in this paragraph are:

• **Color Restore**	**(§ 3.4.1, on 3-18)**
• **Color Select**	**(§ 3.4.2, on 3-19)**
• **Color Map**	**(§ 3.4.3, on 3-20)**
• **Color Transparency**	**(§ 3.4.4, on 3-21)**
• **Upload Transparency**	**(§ 3.4.5, on 3-22)**

3.4.1 Color Restore

Format: `\CR{\color...}\\`

Parameters: `color` = color index (1 - 8) to restore (optional)

Return: none

Description:

The Color Restore command is used to restore the default DigiBox CODI color for the color index specified. If no color parameters are specified, DigiBox CODI restores all 8 colors to their original palette values.

3.4.2 Color Select

Format: `\Cn\\`

Parameters: `n` = color index (1 - 8) (default = `\C1\\`)

Return: none

Description:

The Color Select command allows you to make one of the eight available colors active. After a color has been selected, all characters sent to the DigiBox CODI are displayed in that color. The default color at start-up is 1 (white).

3.4.3 Color Map

Format: `\CM\color1\color2\\`

Parameters:

`color1` = start color index (1 - 7)
`color2` = end color index (2 - 8)

Return: none

Description:

The Color Map command is used to define which colors in the palette will be used for a multi-colored logo. The parameter ***color1*** identifies the start color index in the DigiBox CODI palette in which to store the logo colors. The parameter ***color2*** identifies the end palette entry. The maximum number of colors in a logo is seven. For example, if a logo contained 3 colors and it was to be displayed starting with the color 5, the command would be **`\CM\5\8\\`.**

NOTE

Multi color logos are no longer supported by Chyron since they have been replaced *tiff, Targa or Chyron* RGBA characters. This command is retained for use with existing logos.

Before you can use a logo, the following steps must be performed:

[ ] 1. Download the logo file using the Download Font command.

[ ] 2. Create the logo colors using the Download, Palette or Color Setup command.

[ ] 3. Use the Color Transparency command to specify the logo transparency level. You must set the color body AND edge color to the same transparency level.

[ ] 4. Use the Color Map command to specify the character colors to be used for the logo.

NOTE

Set up the colors *before* you specify the color mapping. If you set up the colors after mapping, you must redo the color mapping. The colors you map for multi-colored logos are no longer available for characters. You should not display characters using the logo colors. Two or more logos can use the same logo colors.

3.4.4 Color Transparency

Format: `\CT\color\type\percent{\color\type\percent...}\\`

Parameters: `color` = color index (1-9; 9 = colors 1-8 effected)

`type` = type of transparency (1,2):

`1` = character body
`2` = character edge

`percent` = amount of color which remains visible (0 - 100): 0 = blank, 100 = Full On

Return: none

Description:

The Color Transparency command allows you to change the transparency level of one or more colors. This command allows you to specify the color to be made transparent, the character body or edge, and the percentage of the color to remain visible. By specifying color nine (9), you can make the transparency level of all eight colors the same in one command.

3.4.5 Upload Transparency

Format: `\UR\index\\`

Parameters: `index` = color transparency index (1-9):
`1-8` = upload transparency for this color index
`9` = upload transparency for all colors

Return: `\CT\color\1\body\2\edge\\`
`color` = color index (1 - 8)
`body` = percent body transparency (0 - 100)
`edge` = percent edge transparency (0 - 100)

Description:

The Upload Transparency command uploads DigiBox CODI color transparency information for the specified color to the host. The system provides both the character body and edge transparency percentage. The percentage refers to the amount of color remaining visible. A level of 100% refers to a full-on color. By specifying mode nine (9), the system responds with the transparency level of all eight system colors.

3.5 Display Commands

DigiBox CODI utilizes display commands to copy (store) display information on a fixed disk file and to control specific display characteristics.

The commands that are described in this paragraph are:

• **Upload Display**	**(§ 3.5.1, on 3-24)**
• **Fade Rectangle**	**(§ 3.5.2, on 3-25)**
• **Dissolve Rectangle**	**(§ 3.5.3, on 3-26)**
• **Bit Blitting**	**(§ 3.5.4, on 3-27)**

3.5.1 Upload Display

Format: `\UD\mode\X position\Y position\Height\Width\[filename]\\`

Parameters:
`mode` = `0` .bmp - Bitmap format
`mode` = `1` .chy - Chyron RGBA format
`mode` = `2` .tif - TIF format
`mode` = `3` .tga - TARGA format

`X position, Y position` = the starting coordinates of the display at which the upload should begin. Coordinates 0,0 denote the upper left corner of the monitor.

X-range = 0 - 719
Y-range = NTSC: 0 - 485, PAL: 0 - 575

`height` = the height in scanlines of the upload area
(NTSC: 1 - 485, PAL: 1 - 585)

`width` = the width in pixels of the upload area
(1 - 720)

`[filename]` = the path and name of the file that will hold the graphic

Return:
File not Found
Invalid Parameter

Description:

The Upload Display command is used to copy a display bitmap onto a disk file in the available formats determined by the operating mode.

3.5.2 Fade Rectangle

Format: `\MF\mode\X position\Y position\Height\Width\Frames\\`

Parameters:

`mode` = 0 Fade Up from black
`mode` = 1 Fade Down from All On

`X position, Y position` = the starting coordinates of the display at which the fade should begin. Coordinates 0,0 denote the upper left corner of the monitor

X-range: (0-719)

Y-range (NTSC: 0-485, PAL: 0-575)

`heigh`t = height in scanlines of the fade area (NTSC: 1 - 485, PAL 1 - 585, 16 scanline increments)

`width` = width in pixels of the fade area (1 - 720)

`frames` = number of frames for the fade effect (1 - 255)

Return: none

Description:

The Fade Rectangle Command is used to “fade” a rectangle of pixels on the screen on and off at a specific rate.

3.5.3 Dissolve Rectangle

Format: `\MV\X position\Y position\Height\Width\Frames\\`

Parameters:

`X position, Y position` = the starting coordinates of the display at which the dissolve should begin. Coordinates 0,0 denote the upper left corner of the monitor

X - range: (0-719)
Y - range (NTSC: 0-485, PAL: 0-575)

`height` = the height in scanlines of the dissolve area (NTSC 1 - 485, PAL 1 - 585, 8 scanline increments)

`width` = the width in pixels of the dissolve area (1 - 720)

`frames` = the number of frames for the dissolve effect (1 - 255)

Return: none

Description:

The Dissolve Rectangle Command is used to produce a "Dissolve Effect" between the Displaying and the Non-Displaying Buffers. This causes the non-displaying buffer to fade up as the displaying buffer fades down.

3.5.4 Bit Blitting - Function Call

Format: `BltPixels(INT32 iBoard, int iStartX, int iStartY, int iHeight, int iWidth, unsigned int *pBuf, int iDisplay), int iDirection;`

Parameters:

`INT32 iBoard` = The number of the digital- pcCODI board (0 - 7).

`int iStartX` = Specifies the Starting Pixel X-address on the scanline. The specified address can be any value from a negative number to 719.

`int iStartY` = Specifies the Starting scanline Y-address on the raster. The specified address can be any value from a negative number to 624.

`int iHeight` = Specifies the height of the buffer passed in scanlines, usual range is 1 - 625.

`int iWidth` = Specifies the width of the buffer passed in pixels, usual range is 1 - 720.

`unsigned int *pBuf` = This is a pointer specifying the start of the buffer that holds the pixels to be copied. Each pixel is represented by an RGBA component as follows:

Red = Byte 0
Green = Byte 1
Blue = Byte 2
Alpha = Byte 3

NOTE

The pixel values must be prescaled in order to display valid video. Therefore, no *Red, Green* or *Blue* Byte can be larger than the *Alpha* component.

For example, if the Alpha component is set to 0x3f, no Red/Green/Blue value can be greater than 03xf.

`int iDisplay` = Enables blitting into the displaying or non-displaying buffer by setting this value to 1 or 0 before the Blit operation:

`0` = Draw into non-displaying buffer
`1` = Draw into displaying buffer

NOTE

If it is required to Blit into the non-displaying buffer, a Message effects command with a -1 as the message id can be used to bring that contents of that buffer into the display.

`int iDirection` = Determines status of DigiBox CODI frame buffer memory (`0` or `1`):

`0` = Enables drawing to DigiBox CODI frame buffer memory

`1` = Enables copying the contents of the DigiBox CODI frame buffer memory to a local memory location

NOTE

Make absolutely sure that enough space is allocated in the local memory buffer to hold the image that is being copied. Otherwise the result will be the, *"Blue Screen of Death."*

Return: None

Description:

This function can be used to copy a rectangular block of pixels, pointed to by the **`*pBuf command,`** directly into the displaying frame buffer of the DigiBox CODI.

3.6 Event Timer Commands

The Event Timer allows you to display a count up/down timer in any one of 21 different display formats. Formats include displaying minutes or seconds to three whole places, and displaying the hundredths place when the timer is not being updated on the display. In addition, each format may be displayed with leading zeros or leading spaces, and each field may be displayed using any font or color.

CAUTION

The Event Timer Commands are intended only for display purposes. They are not intended to be used for precise timing measurements.

The commands that are described in this paragraph are:

• **Event Timer Index**	**(§ 3.6.1, on 3-30)**
• **Event Timer Display**	**(§ 3.6.2, on 3-31)**
• **Event Timer Setup**	**(§ 3.6.3, on 3-32)**
• **Event Timer Begin**	**(§ 3.6.4, on 3-34)**
• **Event Timer Laps**	**(§ 3.6.5, on 3-35)**
• **Event Timer Erase**	**(§ 3.6.6, on 3-36)**
• **Event Timer Time**	**(§ 3.6.7, on 3-37)**
• **Event Timer Color**	**(§ 3.6.8, on 3-38)**
• **Event Timer Font**	**(§ 3.6.9, on 3-39)**
• **Event Timer Position**	**(§ 3.6.10, on 3-40)**

The Event Timer Time command allows you to set the start time of the timer. The timer can display hours, minutes, seconds, tenths and hundredths of a second.

The Laps time feature allows you to stop the timer from updating the display while the timer is running in the background. When the laps time is no longer required, the DigiBox CODI can be instructed to continue updating the timer display.

In addition to the 21 display formats, you may create your own display format by repositioning the timer fields to any location on the display. You may add your own titles to describe each of the displayed fields.

3.6.1 Event Timer Index

Format: `\EI\timer\\`

Parameters: `timer` = index to event timer (1-4)

Return: none

Description:

The Event Timer Index command specifies the current event timer. All other event timer commands will operate on the specified event timer until another Event Timer Index command is used. However, the Event Timer Display, Event Timer Begin, Event Timer Laps and Event Timer Erase commands provide optional parameters that can directly address one or more event timers at the same time. If the optional parameter is not specified with these commands, the current event timer index is used.

3.6.2 Event Timer Display

Format: `\ED\mode{\timer...}\\`

Parameters: **`mode`** = display mode (0,1):

`0` = disable the timer display
`1` = enable the timer display

`timer` = index to event timer (1 - 4) (optional)

Return: none

Description:

The Event Timer Display command is used to enable or disable the event timer display. Before the timer is displayed, all setup commands must have been performed. When the timer is enabled, it may or may not be already running. The timer does not begin running until the Event Timer Begin command is issued in mode 1. When the timer display is disabled, it may or may not be running. The timer does not stop running until the Timer Begin command is issued in mode 0.

If the timer parameter is not specified, this command operates on the current timer as specified with the Event Timer Index command. If one or more timer parameters are specified, this command operates on only those timers. This allows you to display multiple timers simultaneously.

3.6.3 Event Timer Setup

Format: `\ES\xaddr\yaddr\color\font\format\mode{\pad\type}\\`

Parameters: `xaddr` = x-pixel address for timer text 0 - 719

`yaddr` = y-pixel address for timer text (NTSC 0-485, PAL 0-575)

`color` = text color index (1 - 8)

`font` = text font index (1 - 16)

`format` = timer format (text)

:

Table 3-2: Hour/Minute/Second Formats

Hour Formats	Minute Formats	Second Formats
HH:MM:SS.TH	MM:SS.TH	SS.TH
HH:MM:SS.TS*	MM:SS.TS	SS.TS
HH:MM:SS.T	MM:SS.T	SS.T
HH:MM:SS	MM:SS	SS
HH:MM	MMM:SS.TH	SSS.TH
	MMM:SS.TS	SSS.TS
	MMM:SS.T	SSS.T
	MMM:SS	SSS

@ = any format preceded by an 'N' character displays digits numerically justified.

* = .TS is tenths with silent hundredths place displayed only when timer is stopped.

`Mode` = timer count direction (0, 1):

`0` = up counter

`1` = down counter

`pad` = pad format (0, 1): (optional):

`0` = pad with zeros (default

`1` = pad with space

`type` = lead type (0, 1): (optional)

`0` = one lead zero (default)

`1` = not one lead zero

Return: none

Description:

The Event Timer Setup command is used to specify how the event timer will be displayed. Before the timer can be displayed, the Event Timer Setup command must be executed. There are 21 different timer display formats. The timer digits are normally positioned within each field depending on the selected font. By preceding the format text with the letter <N>, the timer digits are numerically justified, from right to left, on the display. The mode can be set for a count up or count down timer.

3.6.4 Event Timer Begin

Format: `\EB\mode{\timer...}\\`

Parameters: `mode` = run mode (0,1):

`0` = stop timer
`1` = begin timer

`timer` = index to event timer (1-4) (optional)

Return: none

Description

The Event Timer Begin command is used to begin or stop the event timer. If the timer has not yet been displayed using the Event Timer Display command, it will begin running without being displayed. If the timer is displaying the laps time, the current time is updated on the display. If the timer parameter is not specified, this command operates on the current timer as specified with the Event Timer Index command. If one or more timer parameters are specified, this command operates on only those timers. This allows you to start or stop multiple timers at the same time.

It is important that you enter every timer that should start or stop in one Event Timer Begin command. In this way, all timers listed will function simultaneously. By using *successive* commands, no two timers will react simultaneously because of the time it takes DigiBox CODI to interpret the commands.

3.6.5 Event Timer Laps

Format: `\EL{\timer...}\\`

Parameters: `timer` = index to event timer (1 - 4) (optional)

Return: none

Description:

The Event Timer Laps command displays the laps time on the monitor but does not stop counting. Use the Event Timer Begin command to resume updating the display with the counting timer. This command can be used only if the timer is displayed and running.

If the timer parameter is not specified, this command operates on the current timer as indicated by the Event Timer Index command. If one or more timer parameters are specified, this command operates on only those timers. This allows you to display the laps time of multiple timers together.

3.6.6 Event Timer Erase

Format: `\EE{\timer...}\\`

Parameters: `timer` = index to event timer (1-4) (optional)

Return: none

Description:

The Event Timer Erase command is used to stop and erase the event timer.

If the timer parameter is not specified, this command operates on the current timer as indicated by the Event Timer Index command. If one or more timer parameters are specified, this command operates on only those timers. This allows you to stop and erase multiple timers simultaneously.

3.6.7 Event Timer Time

Format: `\ET\hhmmss{th}\\` (format 1) `or`

`\ET\hhmmmsssth\\` (format 2)

Parameters: format 1:

`hhmmssth` = timer start time to set:
`hh` = hour (00 - 23)
`mmm` = minutes (00 - 59)
`sss` = seconds (00 - 59)
`th` = hundredths of a second (00 - 99) (optional)

format 2:

`hhmmssth` = timer start time to set:
`hh` = hour (00 - 23)
`mmm` = minutes (000 - 999)
`sss` = seconds (000 - 999)
`th` = hundredths of a second (00 - 99) (mandatory)

Return: none

Description:

The Event Timer Time command sets the time for the DigiBox CODI event timer. If the timer tenths and hundredths places are not specified, both are assumed to be zero. In format 1, the hundredths of a second parameter (`th`) is optional. In format 2, the parameter is mandatory.

NOTE

Optionally, if an E is placed before the hours (hh) field this time will be used as the END time to stop the timer. For example, if you wish the timer to start at hours:00, minutes:00, seconds:00 and tenths:00 you would enter `\ET\00000000\\`. If you wish the time to end at hours :00, minutes :05, seconds :12 and hundredths :53 you would enter `\ET\E00051253\\`.

3.6.8 Event Timer Color

Format: `EC\field\color{\field\color...}\\`

Parameters: `field` = field to effect (H,M,S,T):

`H` = hours
`M` = minutes
`S` = seconds
`T` = tenths\hundredths

`color` = color index (1 - 8) for field

Return: none

Description:

The Event Timer Color command is used to set up the timer color for the specified field. The command should be used after the Event Timer Setup command to change the color attribute of any field. Multiple fields can be changed in a single command.

3.6.9 Event Timer Font

Format: `\EF\field\font{\field\font...}\\`

Parameters: `field` = field to effect (H,M,S,T):

H = hours
M = minutes
S = seconds
T = tenths\hundredths

`font` = font index (1 - 8) for field

Return: none

Description:

The Event Timer Font command is used to set up the font for the specified field. The command should be used after the Event Timer Setup command to change the font attribute of any field. Multiple fields can be changed in a single command.

3.6.10 Event Timer Position

Format: `\EP\field\xaddr\yaddr{\field \xaddr\yaddr...}\\`

Parameters: `field` = field to effect (H,M,S,T):

- `H` = hours
- `M` = minutes
- `S` = seconds
- `T` = tenths\hundredths

`xaddr` = x-pixel address to display field 0 - 719

`yaddr` = y-pixel address to display field (NTSC: 0 - 485, PAL 0 - 575)

Return: none

Description:

The Event Timer Position command is used to set up the X,Y coordinates of a field on the display. Each timer field can be independently positioned anywhere on the display. This command should be used after the Event Timer Setup command to change the x,y coordinates of any field. If this command is used, the field digits are always right-justified and the field delimiter is not displayed. Multiple fields can be changed with a single command.

3.7 File System Commands

There are three new File system Commands available with the DigiBox CODI. these are:

- **Set Message Directory Path** **(§ 3.7.1, on 3-42)**
- **Set Font Directory Path** **(§ 3.7.2, on 3-43)**
- **Set Graphic Directory Path** **(§ 3.7.3, on 3-44)**

3.7.1 Set Message Directory Path

Format: `\MP\{message directory path}\\`

Parameters: `message directory path` = sets the path of the message directory

Return: none

Description:

This command loads the path name to the directory that contains the DigiBox CODI or iNFiNiT! messages. By using this command to set the message directory path, only the file name is required in the Download Message command and not the full file path.

3.7.2 Set Font Directory Path

Format: `\FP\[font directory path]\\`

Parameters: `[font directory path]` = sets the path of the font directory

Return: none

Description:

This command loads the path name to the directory that contains the DigiBox CODI or Infinit! fonts. When this path name is loaded, the full path to the font directory for the Download Message command (`\DM\I`) is not needed.

3.7.3 Set Graphic Directory Path

Format: `\GP\[graphic directory path]\\`

Parameters: `[graphic directory path]` = sets the path of the graphic directory

Return: none

Description:

This command loads the path name to the directory that contains the DigiBox CODI or Infinit! graphics. When this path name is loaded, the full path to the graphic directory for the Download Graphic (**`\DG`**) or Download Background (**`\DB`**) commands is not needed.

3.8 Flash Commands

The Flash commands allows the set up and flashing of specified colors on an as needed basis. The Flash Setup command specifies the flash color on-time and off-time. The Flash Begin command initiates the color flash which continues to flash until the Flash End command is executed.

NOTE

Some Message Effect commands will temporarily suspend the color flash while the transition effect is in progress.

The flash commands that are described in this paragraph are:

- **Flash Setup**	**(§ 3.8.1, on 3-46)**
- **Flash Begin**	**(§ 3.8.2, on 3-47)**
- **Flash Multiple**	**(§ 3.8.3, on 3-48)**
- **Flash End**	**(§ 3.8.4, on 3-49)**

3.8.1 Flash Setup

Format: `\FS\color\on-time\off-time\tocolor`
`\percent{\color\on-time\off-time{\tocolor\percent...}}\\`

Parameters: `color` = color index (1-8) to flash

`on-time` = time color is on (NTSC: 60/sec,
PAL: 50/sec)

`off-time` = time color is off (NTSC: 60/sec,
PAL: 50/sec)
`tocolor` = color index (1 - 8) displayed during off-time

`percent` = percentage of tocolor displayed (0 - 100)

Return: none

Description:

The Flash Setup command allows you to set up a color to flash. The length of the flash is defined by the duration of the on-time and off-time of the flash color. In addition, you can specify the display color and transparency percent during the flash off-time. By specifying a zero percentage, nothing is displayed during the flash off-time. If 100 percent is specified, the full intensity of the tocolor is displayed during the off-time. The tocolor and percent parameters are optional fields and are used only for the last flash setup color. If these parameters are not displayed, tocolor=color and percent=0.

NOTE

If the Color Transparency command is used to set a transparency level for the tocolor. The percent parameter of the Flash Setup command will be a percentage of that value.

3.8.2 Flash Begin

Format: `\FB\\`

Parameters: none

Return: none

Description:

The Flash Begin command flashes the last color set using the Flash Setup command.

3.8.3 Flash Multiple

Format: `\FM{\color...}\\`

Parameters: `color` = color index (1 - 8) to flash (optional)

Return: none

Description:

For Legacy reasons, the Flash Multiple command has been retained. When used, it will only begin flashing the color set by the Flash Setup command, regardless of the color parameters that have been set.

3.8.4 Flash End

Format: `\FE{\color...}\\`

Parameters: `color` = color index (1 - 8) to end flash (optional)

Return: none

Description:

The Flash End command stops the color from flashing.

3.9 Font Commands

DigiBox CODI supports up to sixteen (16) fonts simutaneously. These fonts can be the machine fonts created on the Infinit family of Chyron equipment. Conversely, the fonts can be Codi machine fonts created from Window true type fonts using the ***Build Font*** command.

DigiBox CODI fonts are sized by scanlines. The height of a font is measured from the top of the capital letter "A" to the bottom of the lower case "g." Therefore, if you were to use a 20-scanline font with capital letters only, the characters would in fact be smaller than 20 scanlines.

Machine fonts can be loaded on the DigiBox CODI using the ***Download Font*** command.

There are four types of edges available for DigiBox CODI fonts: drop shadow, character offset, border or none. The DigiBox CODI does not add edges dynamically to a character as they are being displayed. They must be part of the font when the font is downloaded. However, a font edge can be made completely transparent using the Color Transparency command.

The commands that are described in this paragraph are:

- **Download Font** **(§ 3.9.1, on 3-51)**
- **Download iNFiNiT! font** **(§ 3.9.2, on 3-52)**
- **Font Select** **(§ 3.9.3, on 3-53)**
- **System Font** **(§ 3.9.4, on 3-54)**
- **Download Virtual Font** **(§ 3.9.5, on 3-55)**

3.9.1 Download Font

Format: `\DF\font\\data` or `\DF\font\[filename]\\`

Parameters: `font` = font ID 1 - 16

`data` = binary font data

`[filename]` = path and filename of font to be loaded

Return: none

Description:

The ***Download Font*** command is used to load either an Infinit or Codi machine font.

1. `\DF\font\\data`

This form of the ***Download Font*** command allows you to download a new font to the specified font index. In order to use a font that is different from the default DigiBox CODI fonts, it must be downloaded. It is suggested that the Acknowledge mode be set before downloading font data so that the host can determine when the DigiBox CODI has completed processing the downloaded data.

The Acknowledge mode is set by using the System Communication Mode command (§ 3.19.8, on 3-148). If an error occurs while downloading the font, DigiBox CODI restores the original default font. This command uses a portion of the available memory within the DigiBox CODI to save the font data.

2. `\DF\font\[filename]\\`

This form of the ***Download Font*** command, is the preferred method of loading fonts. Virtual font technology is only available when fonts are loaded using the form of the ***Download Font*** command.

When using this command, the filename of the machine font to be loaded must be enclosed in brackets ({ }). If the font directory was previously set with the Set Font Directory command, then only the machine font filename needs to be used, otherwise the full Windows path name is recommended.

3.9.2 Download iNFiNiT! Font

Format: `\DI\font\{filename}\\`

Parameters: `font` = font ID 0 - 15

`filename` = path and filename of iNFiNiT! font to be loaded

Return: none

Description:

Enable iNFiNiT! font data to be loaded into the DigiBox CODI directly from the Windows file.

3.9.3 Font Select

Format: `\Fn\\`

Parameters: `n` = font index (1 - 16, default = `\F1\\`)

Return: none

Description:

The Font Select command allows you to make one of the sixteen system fonts active. After selecting a font, typed characters are displayed in that font until a new Font Select command is executed. If the font index specified is not loaded, the system responds with an error message. Font 1 is assigned as the default font.

3.9.4 System Font

Format: `\SF\\`

Parameters: none

Return: `Font n: name, Height: h, Edge: e, Size: s, File Name: fn, File Size: bytes`

`n` = font index (1-16)

`name` = font name

`h` = font height

`e` = edge type (None, Shadow, Border, Offset)

`s` = edge size

`fn` = file name of font loaded from Font Files

`bytes` = size in bytes of file loaded from Font Files

Description:

The System Font command causes DigiBox CODI to respond with a list of the fonts that are currently installed. Each font is described by its font index, font name, height, edge style and edge size. If a font index is not loaded, the system displays a "Not Loaded" message. If a font index is not loaded but contains one or more downloaded PCX graphic images, the system responds with a "Graphics Loaded" message.

The System Font command include two additional parameters for fonts that were loaded directly from Font Files. These are file name and file size.

3.9.5 Download Virtual Font

Format: `\DF\font\V\{filename}\\`

Parameters: `font` = Font ID number, 0 - 15

`V` = Specifies virtual font

`filename` = Path and filename of font to be loaded

Return: none

Description:

The parameter **`V`** specifies that this font is to be loaded as a Virtual Font. By loading as a Virtual Font, a minimum amount of the on board memory is used. Normally the contents of the entire font are loaded into the pcCODI memory. When a font is loaded with the **`V`** parameter then only the characters that are displayed on the pcCODI output are loaded into memory, not the whole font. This better manages the use of the available on-board memory.

3.10 GPIO Commands

There are two types of General Purpose Input/Output (GPIO) commands, Internal GPIO commands and External GPIO commands.

Internal GPIO commands are used to set up and execute a sequence of ASCII codes that can represent Codi commands or simply characters to display. There can be up to 1000 characters per GPI buffer. An Internal GPI event can be triggered by the following events:

- A clock event triggered when a pre-set time occurs. (The clock time is set by using the GPI Time command.)
- The host using the GPI Begin command.

The Internal GPIO commands that are described in this paragraph are:

• **GPI Setup**	**(§ 3.10.1, on 3-57)**
• **GPI Begin**	**(§ 3.10.2, on 3-58)**
• **GPI Time**	**(§ 3.10.3, on 3-59)**
• **GPI Time Reset**	**(§ 3.10.4, on 3-60)**

External GPIO commands are events that are controlled by or themselves control external devices. Control of external devices is accomplished through specific pins on the DigiBox CODI board. External GPIO commands operate in a different manner than Internal GPIO commands since there is more interaction with the host application.

The External GPIO commands that are described in this paragraph are:

• **GPI Output**	**(§ 3.10.5, on 3-61)**
• **GPI Input**	**(§ 3.10.6, on 3-62)**
• **Register GPIO Callback Procedure**	**(§ 3.10.7, on 3-63)**

3.10.1 GPI Setup

Format: `\GS\gpi\data\\`

Parameters: `gpi` = gpi index (1-8)

`data` =PC Codi characters and commands to display for gpi.

`data` parameter *cannot* contain the following commands:

`\ME\CS` = Crawl Setup Command.
`\ME\RS` = Roll Setup Command.
`\DF` = Download Font Command.
`\DG` = Download Graphic Command.

Return: none

Description:

The GPI Setup command configures the PC Codi for the execution of a GPI event. The GPI setup data can consist of any combination of characters and commands *exclusive of the following commands:*

- Crawl Setup
- Roll Setup
- Download Font
- Download Graphic

The total length of the GPI setup data is 1000 characters for each GPI.

3.10.2 GPI Begin

Format: `\GB\gpi\\`

Parameters: `gpi` = gpi index (1 - 8)

Return: none

Description:

The GPI Begin command simulates the activation of a remote GPI. By using this command, the DigiBox CODI will execute the GPI commands as if a remote GPI had actually occurred.

3.10.3 GPI Time

Format: `\GT\gpi\hhmmss{\interval}\\`

Parameters: **`gpi`** = gpi index (1-8)

`hhmmss` = gpi trigger time (24-hour mode):

`hh` = hour
`mm` = minute
`ss` = second
`interval` = interval to next gpi trigger time (hhmmss format) (optional)

Return: none

Description:

The GPI Time command allows you to set the time at which a GPI should be triggered. Use the GPI Setup \GS command to set up the DigiBox CODI commands that should be executed when the GPI time is reached. This function does not require that the clock be displayed in order for the GPI to be triggered.

Optionally, you may want to specify a time interval. This option allows you to perform a GPI trigger at each interval starting with the trigger time.

3.10.4 GPI Time Reset

Format: `\GR\gpi\\`

Parameters: `gpi` = gpi index (1 - 8)

Return: none

Description:

The GPI Time Reset command resets the GPI trigger time.

3.10.5 GPI Output

Format: `\GO\gpi\flag\\`

Parameters: `gpi` = GPIO port number (0 - 15)

`flag` = 0 or 1

Return: none

Description:

The GPI Output command transmits a 0 or 1 signal level through the selected GPIO port. This command is used to turn on a light or to signal another piece of equipment of an event.

NOTE

The external GPIO devices can all be Input, all Output or a combination of both (Input/Output). There are a total of 16 GPIO ports. Therefore, even though the GPI range can be 0 - 15 on both the `\GO` (output) or `\GI` (input) commands, only a total of 16, not 32, ports are available.

3.10.6 GPI Input

Format: `\GI\gpi\\`

Parameters: `gpi` = GPIO port number (0 - 15)

Return: 0 or 1

Description:

When issued, the GPI Input command reads the status of the specified GPIO port number and returns a 0 or 1 signal level to the source.

NOTE

The external GPIO devices can all be Input, all Output or a combination of both (Input/Output). There are a total of 16 GPIO ports. Therefore, even though the GPI range can be 0 - 15 on both the `\GO` (output) or `\GI` (input) commands, only a total of 16, not 32, ports are available.

3.10.7 Register GPIO Callback Procedure

Format: `RegisterGPIOCCallback(int pcCODI board, HWND, Message int GPIO Index, int contact)`

Parameters: `pcCODI board` = DigiBox CODI index starting at 0

`HWND` = handle to this application's main window

`Message` = user-defined message that will be placed on the Windows Message Queue. This connects the function to the call from the pcCODI GPIO process.

`GPIO index` = GPIO pin index (0 - 15)

`contact:` `0` = call when GPIO is changed to 0
`1` = call when GPIO is changed to 1
`2` = call each time the GPIO changes from 0 to 1 or from 1 to 0

`CallBackFunction protocol:`

CallBackFunction(WPARAM wParam, LPARAM lParam); The State of the contact is passed in wParam; lParam = 0.

Return: none

Description:

The GPIO processing circuitry enables an application to register a Callback function with a specific GPIO. A Callback is a function in an application that can be called when the GPIO status changes. For example, a GPIO pin is connected to a switch like a keyboard switch. Each time the switch is pressed or released, or both, that function will be called. The Register GPIO Callback Procedure allows a record to be made of each action.

3.11 Graphic Commands

DigiBox CODI can display PCX, TARGA, TIFF, BMP and Chyron RGBA graphic images. The commands that are described in this paragraph are:

- **Download Graphic**	**(§ 3.11.1, on 3-65)**
- **Download Pixels**	**(§ 3.11.2, on 3-67)**
- **Graphic Erase**	**(§ 3.11.3, on 3-68)**
- **Graphic Transparency**	**(§ 3.11.4, on 3-69)**
- **Upload Pixels**	**(§ 3.11.5, on 3-70)**

3.11.1 Download Graphic

Format: `\DG\type\font\key\spalette\npalette{\red\green\blue}\\data`

or

`\DG\type\font\key\spalette\npalette{\red\green\blue}[filename]\\`

Parameters: **`type`** = graphic file type:

`1` = PCX
`2` = 8 bit TGA
`3` = Chyron RGBA
`4` = 24/32 bit TGA
`5` = Tiff
`6` = BMP

`font` = font index (1 - 16) in which the graphic is to be saved

`key` = any keyboard key with the exception of the <Enter> key and the <DEL> key

`spalette` = Not used.

`npalette` = Not used

`red` = Not used

`green` = Not used

`blue` = Not used

`data` = graphic binary data

`[filename]` =path and filename of graphic file to be loaded

Return: none

Description:

The Download Graphic command allows you to download a graphic image and save it as a member of an existing font which can be displayed by selecting that font and displaying that character. The image is stored as a key code in the specified font.

After the PCX graphic image has been downloaded, it can be displayed by sending the specified key code from the specified font. An optional RGB key can be specified in the command which represents an area of the PCX graphic that is transparent.

It is suggested that the Acknowledge mode be set before downloading graphic data so the host can determine when the DigiBox CODI has completed processing downloaded data.

The Acknowledge mode is set by using the System Mode command. This command uses a portion of the available memory within the DigiBox CODI to save the graphic image data.

3.11.2 Download Pixels

Format: `\DX\Xposition\position\Height\Width\Transparency\mode\\data`

Parameters: `X position` = the starting pixel column in the frame buffer for the bitmap (0 - 719)

`Y position` = the starting scanline in the frame buffer for the bitmap (0 = top of screen, NTSC: 0 - 485, PAL: 0 - 575)

`Height` = the height in scanlines of the bitmap

`Width` = the width in pixels of the bitmap

`Transparency` = range of 0 - 255

.

`mode` = 0 (copy operation), 1 (blend operation)

`data` = bitmap being transferred

Return: none

Description:

The Download Pixels command will write the raw pixels directly into the foreground frame buffer. These are raw pixel data. The byte order is: 8 bits BLUE, 8 bits GREEN, 8 bits RED, and 8 bits ALPHA. The pixel data must be pre-shaped video. That means that each of the red, green or blue 8-bit values must **NEVER** be greater than the 8-bit **ALPHA** value. Thus, a pixel of RED=255, GREEN=20, BLUE=128 and ALPHA=150 would yield an invalid pixel and would produce an unpredictable pixel image. If you wanted to display this color 255,20,128 with an ALPHA of 150, you would need to multiply each R,G,B value by 150/255, which would then yield 150,12,75.

The *transparency* parameter applies a global transparency value to all of the pixels in the bitmap before they are written to a frame buffer. If a straight copy is to be done, assign this argument a value of 0.

A *mode* value of 0 replaces the pixels in the foreground frame buffer with the pixels in this bitmap. A *mode value* of 1 will blend, or merge, the existing pixels in the frame buffer with the pixels in this bitmap.

The bitmap (`data` parameter) must be transmitted using the WriteMemPCCODI() function and its length in bytes must be exactly: *Height * Width * 4.*

3.11.3 Graphic Erase

Format: `\GE\font\key\\`

Parameters: `font` = font index (1 - 16) to erase graphic

`key` = key in font to use (1 - 255 except 10=<LF>, 13=<CR>, 17 =<XOFF>, 19 =<XON>, 127 =<DEL>)

Return: none

Description:

The Graphic Erase command allows you to erase a graphic image that has been previously downloaded using the Download Graphic command. This command frees the PC Codi memory allocated to save the graphic image data.

3.11.4 Graphic Transparency

Format: `\GL\font\key\percent\\` - for PCX

Parameters: `font` = font index (1 - 16) to erase graphic

`key` = key in font to use (1-192 except 10=<LF>, 13=<CR>, 17=<XOFF>, 19=<XON>, 127=<DEL>)

`percent` = amount of color which remains visible (0 - 100)

Return: none

Description:

The Graphic Transparency command allows you to set the transparency level of a graphic image. When the percent field is set to 100, the graphic is full-on. When set to 0, the graphic is not visible.

3.11.5 Upload Pixels

Format: `\UD\X position\Y position\Height\Width\\`

Parameters: `X position` = the starting pixel column in the frame buffer for the bitmap (0-719)

`Y position` = the starting scanline in the frame buffer for the bitmap:

Top of screen =	0
NTSC =	0 - 485
PAL =	0 - 575

`Height` = the height in scanlines of the bitmap

`Width` = the width in pixels of the bitmap

Return: none

Description:

The Upload Pixels command will send back to the host the pixels in the foreground frame buffer. These pixels are raw pixel data. The byte order is 8 bits BLUE, 8 bits GREEN, 8 bits RED and 8 bits ALPHA. When the command is executed, the bitmap will then be transmitted by the DigiBox CODI over the PCI BUS to your program. Use the READMEMPCCODI() function and make sure that you have a buffer set up that equals: *Height * Width * 4.*

3.12 Letter Commands

The Letter commands allow you to reposition or delete any letter on the display. The System Lock mode affects the functionality of the letter position commands. When the System Lock mode is off, only the letter at the cursor position is affected. When the System Lock mode is on, all characters from the cursor position to the end of the row are affected simultaneously.

The commands that are described in this paragraph are:

- **Letter Blank** **(§ 3.12.1, on 3-72)**
- **Letter Up** **(§ 3.12.2, on 3-73)**
- **Letter Down** **(§ 3.12.3, on 3-74)**
- **Letter Left** **(§ 3.12.4, on 3-75)**
- **Letter Right** **(§ 3.12.5, on 3-76)**
- **Hexadecimal UniCode Characters** **(§ 3.12.6, on 3-77)**

3.12.1 Letter Blank

Format: `\LB\\`

Parameters: none

Return: none

Description:

The Letter Blank command deletes the character at the cursor position.

3.12.2 Letter Up

Format: `\LU{\pixels}\\`

Parameters: `pixels` = pixels to push letter up (optional)

Return: none

Description:

The Letter Up command moves the character at the cursor up by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the letter up one pixel. In the Locked mode, all characters to the right of the cursor are also moved up on the row.

3.12.3 Letter Down

Format: `\LD{\pixels}\\`

Parameters: `pixels` = pixels to push letter down (optional)

Return: none

Description:

The Letter Down command moves the character at the cursor down by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the letter down one pixel. In the Locked mode, all characters to the right of the cursor are also moved down on the row.

3.12.4 Letter Left

Format: `\LL{\pixels}\\`

Parameters: `pixels` = pixels to push letter left (optional)

Return: none

Description:

The Letter Left command moves the character at the cursor left by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the letter one pixel to the left. In the Locked mode, all characters to the right of the cursor are also moved left on the row.

3.12.5 Letter Right

Format: `\LR\{pixels}\\`

Parameters: `pixels` = pixels to push letter right (optional)

Return: none

Description:

The Letter Right command moves the character at the cursor right by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the letter one pixel to the right. In the Locked mode, all characters to the right of the cursor are also moved right on the row.

3.12.6 Hexadecimal UniCode Characters

Format: `\HU\xxxx\...\\`

Parameters: `xxxx` = Four-digit hexadecimal number representing UniCode character.

Return: none

Description:

This command displays the UniCode characters by specifying their 4-digit Hexadecimal values (**`xxxx`**). The actual UniCode values can be displayed by running the "Character Map" program from the Windows NT Accessories menu (Windows 2000 Professional: Start > Programs > Accessories > System Tools). One 4-digit Hexadecimal value in ASCII format is represented for each character separated by a backslash (\).

Each UniCode character must be represented by a 4-digit Hexadecimal number consisting of ASCII characters "0 -F." For example, the Greek letter Delta has a UniCode hexadecimal value of 0394 and is represented by **`\HU\0394\\`**. Further, the TradeMark Symbol in the TrueType Arial Font has a Unicode hexadecimal value of 2122 and is represented by **`\HU\2122\\`**.

3.13 Line Draw Commands

The Line Draw commands allow you to draw lines on the DigiBox CODI display. The drawing edge, color, and line size are completely programmable. You can position the drawing cursor to any position on the display, then draw a line to any other position.

Please note that lines drawn on the display are NOT permanent and cannot be saved with messages when using the Message Save command. Any DigiBox CODI command that redraws the display will erase previously drawn lines. Therefore, use these commands to provide emphasis to displays that have already been composed.

The commands that are described in this paragraph are:

• **Draw Color**	**(§ 3.13.1, on 3-79)**
• **Draw Edge**	**(§ 3.13.2, on 3-80)**
• **Draw Line (To)**	**(§ 3.13.3, on 3-81)**
• **Draw Move (To)**	**(§ 3.13.4, on 3-82)**
• **Draw Size**	**(§ 3.13.5, on 3-83)**

3.13.1 Draw Color

Format: `\XC\{color}\\`

Parameters: `color` = color index (0 - 8) to draw

Return: none

Description:

The Draw Color command specifies the color to be used when drawing. Color 0 is black, or external video.

3.13.2 Draw Edge

Format: `\XE\type\\`

Parameters: `type` = edge type for drawing (N, O, B):

`N` = No Edge
`O` = Offset Edge
`B` = Border Edge

Return: none

Description:

The Draw Edge command sets the drawing edge type. The edge color is specified by the color selected using the Draw Color command.

3.13.3 Draw Line (To)

Format: `\XL\{+,-} xaddr\{+,-} yaddr\\`

Parameters: **`xaddr`** = x-pixel address to draw line 0 - 719

`yaddr` = y-pixel address to draw line (NTSC: 0 - 485;
PAL: 0 - 575)

Return: none

Description:

The Draw Line (To) command draws a line from the current draw cursor position to the specified x,y address, using the current color, size, and edge type. When the *xaddr* or *yaddr* is not preceded by a plus (+) or minus (-) sign, the values are absolute. When a sign is used, the draw-to position is relative to the current draw cursor position.

Please note that lines drawn on the display are NOT permanent, and cannot be saved with messages when using the Message Save command. Any DigiBox CODI command that redraws the display will erase any previously drawn lines.

3.13.4 Draw Move (To)

Format: `\XM\{+,-} xaddr\{+,-} yaddr\\`

Parameters: **`xaddr`** = x-pixel address of draw cursor 0 - 719

`yaddr` = y-pixel address of draw cursor
(NTSC: 0 - 485; PAL: 0 - 575)

Return: none

Description:

The Draw Move (To) command moves the draw cursor to the specified x,y address. When the *xaddr* or *yaddr* parameters are not preceded by a plus (+) or minus (-) sign, the values are absolute. When a sign is used, the draw (to) position is relative to the current draw cursor position.

3.13.5 Draw Size

Format: `\XS\size\\`

Parameters: `size` = size of line (1 - 10)

Return: none

Description:

The Draw Size command sets the line size used in Draw operations. The line is drawn using a circular “brush”, and sets the diameter of the circle. (The circle diameter, in pixels, is twice the Draw Size value.)

3.14 Message Commands

When a message is composed on the DigiBox CODI display, it can be saved on the PC hard disk drive for later use. You may save up to a max of 99,999,999 messages. The number of messages is limited only by the amount of space contained on your hard drive. When required, saved messages can be displayed using any PC Codi Message Effect command.

A saved message can also be uploaded to the host. This operation is available for Legacy reasons and is not recommended as a method of saving pages.

The commands that are described in this paragraph are:

- **Download Message**	**(§ 3.14.1, on 3-85)**
- **Download iNFiNiT! Message**	**(§ 3.14.2, on 3-86)**
- **Download iNFiNiT! Message in Non- Display Buffer**	**(§ 3.14.3, on 3-87)**
- **Message Display**	**(§ 3.14.4, on 3-88)**
- **Message Control**	**(§ 3.14.5, on 3-89)**
- **Upload Message**	**(§ 3.14.6, on 3-90)**
- **Message Number**	**(§ 3.14.7, on 3-93)**

3.14.1 Download Message

Format: `\DM\message\mode{\xaddr\yaddr\color\font\text...}\\`

Parameters: `message` = message index (1-99,999,999) to download

`mode` = download mode (0, 1):
- `0` = text information only
- `1` = text, tab, palette, background info

`xaddr` = x-pixel address of character

`yaddr` = y-pixel address of character

`color` = color index (1 - 8)

`font` = font index (1 - 16)

`text` = message text

Return: none

Description:

The Download Message command allows the user to download a message from the host to a specified message index. The downloaded data always begins with the message text information. This includes the x,y pixel address, color, font and ASCII character information for each character in the message text. In mode 0, only the message text is downloaded. The DigiBox CODI saves the tab, palette and background parameters of the current display with the message. In mode 1, the additional tab, palette, and background parameters are also downloaded. Refer to the Upload Message command for the required message parameter formats.

3.14.2 Download iNFiNiT! Message

Format: `\DM\I\[message filename]\[font directory path]\\`

Parameters: `[message filename]` = the path and filename of the iNFiNiT! message to be read. (If the set message directory name function previously set the message directory path name, then the full message path name is dot required on the message file name.)

`[font directory path]` = the path and filename in which the iNFiNiT! machine fonts used in the message are stored. (If the set font directory function was previously called, this parameter can be omitted.)

Return: `ok`

`File not Found`

Description:

The Download iNFiNiT! Message command reads in an iNFiNiT! message and displays it on the DigiBox CODI. Only standard record messages are valid. Special records and Transform messages are not supported. If a message contains TAB fields, the system will convert them to DigiBox CODI TAB fields, which can then be updated using the DigiBox CODI Tab Update commands.

3.14.3 Download iNFiNiT! Message in Non-Display Mode

Format: `\DM\N\{message filename}\{font directory path}\\`

Parameters: **`N`** = Load message in non-display mode

`message filename` = the path and filename of the iNFiNiT! message to be read. (If the set message directory name function previously set the message directory path name, then the full message path name is dot required on the message file name.)

`font directory path` = the path and filename in which the iNFiNiT! machine fonts used in the message are stored. (If the set font directory function was previously called, this parameter can be omitted.)

Description:

The Download iNFiNiT message in Non-Display Mode command loads an iNFiNiT message in the non-displaying buffer and does not automatically display it. Display of the message can the be controlled by any of the Message Effects commands, Tab Update commands, or the System Display command.

3.14.4 Message Display

Format: `\MD\message\\`

Parameters: `message` = message index (1 - 99999999) to display

NOTE

If message index is specified as 0, the system will display the current message number, then increment the global message number by 1.

Return: none

Description:

Once a message has been saved using the Message Save command, it can be displayed at any time using the Message Display command and specifying the message index. This command updates the DigiBox CODI display with the saved message, color palette, background and tab fields. Specifying a message index of 0 causes the system to display the current message number, then increments the message index by one, as shown below. Assume that there are three message stored in CODI memory.

`\MN\1\\` <-- Set Message Number to 1 *(see MN command)*

`\MD\0\\` <-- Displays Message 1 (value of Message Number), then increments the Message Number

`\MD\0\\` <-- Display Mess. 2, increments Message Number to 3

`\MD\0\\` <-- Display Mess. 3, increments Message Number to 4
`\MD\0\\` <-- Returns Error (no Message number 4), then resets Message Number to 1

Note that the Message Number can be used with any command that requires a message.

3.14.5 Message Control

Format: `\MC\iterations\ME\parameters\time{\ME\parameters\time...}\\`

Parameters:

`iterations` = number of times for script (0=continuous)

`parameters` = parameters for message effect to perform

`time` = pause time between commands (NTSC: 60 counts/second, PAL: 50 counts/second)

Return: none

Description:

It is possible to display a series of saved messages automatically using the Message Control command. This command can only use the Fade, Matrix Wipe and Push message effects. The messages can be displayed for a set number of iterations, or continuously if the iterations specified is zero. A continuous message control is terminated by sending a character or command from the host. Each message effect command is separated by a unique delay time to produce the required effect.

NOTE

If the clock and/or timer is displayed, PC Codi will automatically turn them off for the duration of the effect sequence.

3.14.6 Upload Message

Format: `\UM\message\mode\\`

Parameters: **`message`** = message index (1 - 99999999) to upload

`mode` = upload mode (0, 1):
- **`0`** = text information only
- **`1`** = text, tab, palette, background info

Return:

__1.Text__:

`\DM\message\mode{\xaddr\yaddr\color\font\text...}\\`

`message` = message index (1-99) to upload
`mode` = upload mode (0,1):
- **`0`** = text information only
- **`1`** = text, tab, palette, background info

`xaddr`= x-pixel address of character
`yaddr`= y-pixel address of character
`color`= color index (1 - 8)
`font` = font index (1 - 16)
`text` = row text (max 80 characters)

__2.Tabs__:

`\DT{\tab\xaddr\yaddr\color\font\justification...}\\`

`tab` = tab number (1 - 200) or name (max 11 chars)
`xaddr` = x-pixel address of tab field
`yaddr` = y-pixel address of tab field
`color` = color index (1 - 8)
`font` = font index (1 - 16)
`justification` = tab justify (L,LN,R,RN,C,CN):

- **`L`** = left
- **`LN`** = left numeric
- **`R`** = right
- **`RN`** = right numeric
- **`C`** = center
- **`CN`** = center numeric

3.Palette Colors (8):

```
\DP\color\R\rbs\gbs\bbs\rbe\gbe\bbe\res\ges\bes\ree\gee\bee\rms

\gms\bms\rme\gme\bme\color\R\rbs\gbs\bbs\rbe\gbe\bbe\res\ges

\bes\ree\gee\bee\rms\gms\bms\rme\gme\bme...\\
```

color = color index (1 - 8)
rbs = red body start (0 - 255)
gbs = green body start (0 - 255)
bbs = blue body start (0 - 255)
rbe = red body end (0 - 255)
gbe = green body end (0 - 255)
bbe = blue body end (0 - 255)
res = red edge start (0 - 255)
ges = green edge start (0 - 255)
bes = blue edge start (0 - 255)
ree = red edge end (0 - 255)
gee = green edge end (0 - 255)
bee = blue edge end (0 - 255)
rms = red mix start (0 - 255)
gms = green mix start (0 - 255)
bms = blue mix start (0 - 255)
rme = red mix end (0 - 255)
gme = green mix end (0 - 255)
bme = blue mix end (0 - 255)

4. Background Palette:

```
\DB\7\R\r\g\b\k\r\g\b\k...\\
```

r = red (0 - 255)
g = green (0 - 255)
b = blue (0 - 255)
k = key (0 - 255)

5. Display Mode Flag:

```
\DD\dmode\\
```

dmode = display mode (0 - 65535)

6. Colors(8):

```
\DC\color\R\rbody\redge\gbody\gedge
\bbody\bedge\color\R\rbody
\redge\gbody\gedge\bbody\bedge...\
```

`color` = color index (1 - 8)
`rbody` = red body color (0 - 255)
`redge` = red edge color (0 - 255)
`gbody` = green body color (0 - 255)
`gedge` = green edge color (0 - 255)
`bbody` = blue body color (0 - 255)
`bedge` = blue edge color (0 - 255)

7. Transparency (8):

```
\DL\color\body\edge\color\body
\edge...\\
```

`color` = color index (1 - 8)
`body` = body transparency level (0 - 255)
`edge` = edge transparency level (0 - 255)

Description:

The Upload Message command sends the message text information for a specified message to the host. The DigiBox CODI provides the x,y pixel address, color, font and ASCII character information for each character in the message text. When using mode 0, only the message text is uploaded. When using mode 1 to upload all message information, DigiBox CODI also uploads tab, palette, background and transparency data.

3.14.7 Message Number

Format: `\MN\number\\`

Parameters: `number` = global Message Number (range: 1 - 99999999)

Return: none

Description:

The Message Number command allows you to set the system variable message number to any number between 1 and 99,999,999. This command is useful when selecting a pre-set range of messages for automatic display, etc.

For example, you can have a range of ten messages beginning with message number 50, and another ten messages beginning with message number 75. To display the series of messages, the following commands must be sent:

`\MN\50\\` - sets global message index to 50

`\MD\0\\` - displays the first message (#50), then increments

`\MD\0\\` - displays next message (#51), then increments *(repeats the* `\MD\0\\` *command for all 10 messages).*

When the `\MD\0\\` command reaches the end of the valid message range, an error message is generated, and the global Message Number is reset to 1.

To display the next series of messages, the following commands must be sent:

`\MN\75\\` - sets global message index to 75

`\MD\0\\` - displays first message (#75), then increments

`\MD\0\\` - displays next message (#76), then increments (repeats this command for all 10 messages)

.

3.15 Message Effect Commands

DigiBox CODI provides several message transition effects. These effects replace the current message display with a previously saved message using a display effect. They can also operate between the displaying frame buffer and the non-displaying frame buffer.

Message Effect commands operate on the Foreground Frame Buffer, not in the Background Frame Buffer.

In addition, DigiBox CODI supports a Crawl and Roll effect. A Crawl shifts a row of text from right-to-left across the display at any y-pixel address. A Roll shifts any number of rows of text from bottom to top on the display. Both Crawl and Roll support imbedded font and color changes, and have a feature to pause the effect for a specified time. The commands that are described in this paragraph are:

- **Message Effect Cover** **(§ 3.15.1, on 3-95)**
- **Message Effect Hide** **(§ 3.15.2, on 3-96)**
- **Message Effect Peel** **(§ 3.15.3, on 3-97)**
- **Message Effect Pour** **(§ 3.15.4, on 3-98)**
- **Message Effect Reveal** **(§ 3.15.5, on 3-99)**
- **Message Effect Split** **(§ 3.15.6, on 3-100)**
- **Message Effect Weave** **(§ 3.15.7, on 3-101)**
- **Message Effect Crawl Setup** **(§ 3.15.8, on 3-102)**
- **Message Effect Crawl Begin** **(§ 3.15.9, on 3-103)**
- **Message Effect Fade** **(§ 3.15.10, on 3-105)**
- **Message Effect Dissolve** **(§ 3.15.11, on 3-106)**
- **Message Effect Matrix Wipe** **(§ 3.15.12, on 3-107)**
- **Message Effect Push** **(§ 3.15.13, on 3-109)**
- **Message Effect Roll Setup** **(§ 3.15.14, on 3-110)**
- **Message Effect Roll Begin** **(§ 3.15.15, on 3-111)**
- **Message Effect Roll Erase** **(§ 3.15.16, on 3-112)**
- **Message Effect - iNFiNiT! Messages** **(§ 3.15.17, on 3-113)**

3.15.1 Message Effect Cover

Format: `\ME\CO\message\direction\speed{\mode}\\`

Parameters: **`message:`**

`message index =` (1 - 99999999)
`-1 =` (between display and non-display foreground buffers)

`direction` = cover direction (**`U,D`**):

`U` = up
`D` = down

`speed` = cover speed, 1 - 10 (1 = slowest, 10 = fastest)

`mode` = ease mode (**`S,E,SE`**): (optional)

`S` = ease in at beginning of cover
`E` = ease out at end of cover
`SE` = ease at beginning and end of cover

Return: none

Description:

The Cover command covers the current message with the specified message. The Cover Effect folds the new page over the current page.

NOTE

If the clock and/or timer is displayed, DigiBox CODI will automatically turn them off for the duration of the effect.

3.15.2 Message Effect Hide

Format: `\ME\HI\message\direction\speed{\mode}\\`

Parameters: **`message:`**

`message index` = (1 - 99999999)
`-1` = (between display and non-display frame buffers)

`direction` = hide direction (**`U,D`**):

`U` = up
`D` = down

`speed` = hide speed, 1 - 20 (1 = slowest, 20 = fastest)

`mode` = ease mode (**`S,E,SE`**): (optional)
`S` = ease in at beginning of hide
`E` = ease out at end of hide
`SE` = ease at beginning and end of hide

Return: none

Description:

The Hide command slides the specified message over the current message. The optional mode field can be used to accelerate the hide from a stop to the specified speed and/or decelerate the hide from full speed to a stop.

NOTE

If the clock and/or timer is displayed, DigiBox CODI will automatically turn them off for the duration of the effect.

3.15.3 Message Effect Peel

Format: `\ME\PE\message\direction\speed{\mode}\\`

Parameters: **`message:`**

`message index` = (1 - 99999999)
`-1` = (between display and non-display buffers)

`direction` = peel direction (**`U,D`**):

`U` = up

`D` = down

`speed` = peel speed 1 - 10, (1 = slowest, 10 = fastest)

`mode` = ease mode (**`S,E,SE`**): (optional)

`S` = ease in at beginning of peel
`E` = ease out at end of peel
`SE` = ease at beginning and end of peel

Return: none

Description:

The Peel command peels the current message from the display to reveal the specified message. The Peel Effect folds the old message off the screen, thereby revealing the new message.

NOTE

If the clock and/or timer is displayed, DigiBox CODI will automatically turn them off for the duration of the effect.

3.15.4 Message Effect Pour

Format: `\ME\PO\message\speed\\`

Parameters: `message` = message index (1 - 99999999)

`speed` = pour speed 1 - 10, (1 = slowest, 10 = fastest)

Return: none

Description:

The Pour command pours the specified message onto the display. The Pour Effect blanks the foreground and starts to draw each scanline of the new message from the bottom of the screen to the top, then replicates the currently drawn line from the top to the current scanline position.

3.15.5 Message Effect Reveal

Format: `\ME\RV\message\direction\speed{\mode}\\`

Parameters: `message:`

`message index` = (1 - 99999999)
`-1`= (between display and non-display buffers)

`direction` = reveal direction (`U,D`):

`U` = up
`D` = down

`speed` = reveal speed, 1- 20 (1 = slowest, 20 = fastest)

`mode` = ease mode (`S,E,SE`): (optional)

`S` = ease in at beginning of reveal
`E` = ease out at end of reveal
`SE` = ease at beginning and end of reveal

Return: none

Description:

The Reveal command slides the current message off the display to reveal the specified message. The optional mode field can be used to accelerate the reveal from a stop to the specified speed and/or decelerate the reveal from full speed to a stop.

NOTE

If the clock and/or timer is displayed, DigiBox CODI will automatically turn them off for the duration of the effect.

3.15.6 Message Effect Split

Format: `\ME\SP\message\direction\speed{\mode}\\`

Parameters: **`message:`**

`message index` = (1 - 99999999)

`-1`= (between display and non-display buffers)

`direction` = split direction (**`U,D`**):

`U` = center to up/down

`D` = up/down to center

`speed` = split speed, 1 - 10 (1 = slowest, 10 = fastest)

`mode` = ease mode (**`S,E,SE`**): (optional)

`S` = ease in at beginning of split

`E` = ease out at end of split

`SE` = ease at beginning and end of split

Return: none

Description:

The Split command splits the current message horizontally at the center of the display to reveal the specified message.

NOTE

If the clock and/or timer is displayed, DigiBox CODI will automatically turn them off for the duration of the effect.

3.15.7 Message Effect Weave

Format: `\ME\WE\message\ysize\speed{\mode}\\`

Parameters: `message:`

`message index` = (1 - 99999999)

`-1`= (between display and non-display buffers)

`ysize` = size of the row to weave(NTSC 8-242, PAL 8- 286)

`speed` = weave speed, 1-20 (1 = slowest, 20 = fastest)

`mode` = ease mode (`S,E,SE`): (optional)

`S` = ease in at beginning of weave
`E` = ease out at end of weave
`SE` = ease at beginning and end of weave

Return: none

Description:

The Weave command slides alternating rows at the ysize specified from the left and right sides of the display and pushes the current message out and the new message in the display. The optional mode field can be used to accelerate the weave from a stop to the specified speed and/or decelerate the weave from full speed to a stop.

NOTE

If the clock and/or timer is displayed, DigiBox CODI will automatically turn them off for the duration of the effect.

3.15.8 Message Effect Crawl Setup

Format: `\ME\CS\data\\`

Parameters: `data` = text to crawl. `Data` command can also contain:

Font Select command `\Fn\\`
Color Select command `\Cn\\`
System Wait command `\SW\n\\`

where `n` is wait time (NTSC 60/sec, PAL 50/sec)

Return: none

Description:

The Crawl Setup command configures the DigiBox CODI text and commands that will be used during a crawl. Data that can be used in the Crawl Setup command are:

- **Characters**
- **Font Select commands (`\Fn\\`)**
- **Color Select commands (`\Cn\\`)**
- **System Wait commands (`\SW\n\\`)**

Any other DigiBox CODI commands that are imbedded in the crawl data will generate an error response and will be ignored. The crawl data can contain up to 4500 characters. This command uses a portion of the available memory within the DigiBox CODI to save the crawl data.

3.15.9 Message Effect Crawl Begin

Format:

```
\ME\CB\speed\iterations\yaddr{\xStart\width}\\
                                          or
\ME\CB\speed\iterations\yaddr{,yHeight}{\xStart\width}\\
```

Parameters: `speed` = crawl speed (0 - 20):

NOTE

Speed is the number of pixels to shift left per vertical retrace period.

`iterations` = number of crawls to perform (0 = continuous, H = host data)

`yaddr` = y-pixel address to crawl data (NTSC:0 - 485, PAL; 0 - 575)

`xStart\width` = Designates the actual starting point of text crawl on the display

`,yHeight` = Designates the height of the crawl area

Return: none

Description:

Once the Crawl data has been setup on the DigiBox CODI, the Crawl can begin. The Crawl Begin command instructs DigiBox CODI to shift characters from the right side of the screen to the left. As the characters are shifted off the left side of the screen, new characters are displayed from the right of the screen. The DigiBox CODI displays the complete crawl for as many iterations as specified. If 0 iterations are specified, the crawl continues until a new character or command is received from the host. DigiBox CODI will always complete the last iteration of the crawl when a continuous crawl is terminated.

If the optional **`xStart\width`** parameter is included with the command, the text will start to appear at the **`xStart + width`** position on the screen and will crawl off the screen at the **`xStart`** position. Valid **`xStart`** values can be from 0 thru 719, but there are only 720 pixels on a scanline.

Use of the **`xStart\width`** parameter, enables the crawl mask in the hardware. Any foreground video to the left or right of the crawling area will be blanked when the crawl mask is active. To achieve the crawl function into a logo effect, paint the logo into the background using a download background command (§ 3.3, on 3-5).

The **`yHeight`** parameter is optional and is the designated height of the crawling area. If no **`yHeight`** parameter is entered then the height of the current font selected with the **`\Fx\\`** command will be used to determine the number of scanlines to crawl.

Since the Crawl is only a single line on the screen, carriage return characters in the crawl data are translated into space characters.

If iterations is specified as ‘H’ (Host data), crawl data is supplied directly from the host. The DigiBox CODI *does not* crawl the Crawl Setup data. (This mode does not require that the Crawl Setup command be used before beginning the crawl.) The host crawl data can contain any embedded commands as permitted by the Crawl Setup command. The crawl can be terminated by specifying the ‘\\’ character string.

NOTE

If the crawl data is supplied directly from the host, any data previously set up using the Crawl Setup command is erased. You must reset the Crawl Setup data again if it will be required for a crawl. In addition, you must preselect the largest font to be used in the crawl data by using the Font Select command. The current font specifies the crawl window size when the Crawl Begin command is used. If the clock and/or timer are displayed, you must ensure that they are displayed within the static area. If the clock and/or timer are displayed within the active crawl area, they will crawl with the crawl text.

3.15.10 Message Effect Fade

Format: `\ME\FA\message\speed\\`

Parameters: `message` = message index (1 - 99999999) to fade on

`speed` = fade speed (1 - 10)

`1` = slowest

`10` = fastest

Return: none

Description:

The Message Effect Fade command fades the foreground buffer display at a rate determined by the `speed` parameter.

3.15.11 Message Effect Dissolve

Format: `\ME\DS\message\speed\\` or

`\ME\DS\I\{filename}\speed\\`

Parameters: `message` = message index (1 - 99999999) to dissolve

`speed` = number of video frames

`I` = iNFiNiT! message effect

`filename` = filename of iNFiNiT! message to dissolve.

The machine font directory path must have been set with the `\FP` command before this command is used.

Return: none

Description:

When used, the Message Effect Dissolve command simultaneously fades the displaying foreground frame buffer *down,* bringing the non-displaying foreground frame buffer *up,* over the number of frames specified by the `speed` parameter.

3.15.12 Message Effect Matrix Wipe

Format: `\ME\MW\message\type\xsize\ysize\speed{\color}\\`

Parameters: `message` = message index (1 - 99999999) to wipe

`type` = wipe type to perform (0 - 30, as follows):

0 = random order
1 = top to bottom
2 = middle to top and bottom
3 = top and bottom to middle
4 = left to right
5 = middle to left and right
6 = left and right to middle
7 = spiral-out
8 = left to right with a line
9 = top to bottom with a line
10 = diagonal from top-left to bottom-right
11 = diagonal from top-right to bottom-left
12 = diagonal from bottom-left to top-right
13 = diagonal from bottom-right to top-left
14 = multi-diagonal top-left to bottom-right
15 = multi-diagonal top-right to bottom-left
16 = multi-diagonal bottom-left to top-right
17 = multi-diagonal bottom-right to top-left
18 = snake from top-left to bottom-right
19 = snake from top-left to right-bottom
20 = snake from top-right to bottom-left
21 = snake from top-right to left-bottom
22 = snake from bottom-left to top-right
23 = snake from bottom-left to right-top
24 = snake from bottom-right to top-left
25 = snake from bottom-right to left-top
26 = spiral-in
27 = clockwise radar sweep
28 = counterclockwise radar sweep
29 = horizontal blinds
30 = vertical blinds

`xsize` = x-pixels of wipe block (1 - 720)

`ysize` = y-pixels of wipe block (NTSC 0 - 485, PAL: 0-575)

`speed` = speed of message wipe (1- 10, 1 = slowest)

`color` = color index (1 - 8, line color for wipe types 8, 9 only.)

Return: none

Description:

The Matrix Wipe message effect causes a new message to be displayed over the current message using one of 30 different matrix wipe effects. The Matrix Wipe speed depends on the wipe block size and effect type specified. You will have to experiment with the block size, type and speed to produce the desired effect. The Matrix Wipe message effect uses the background and color palette of the original message while performing the effect.

NOTE

If the clock and/or timer is displayed, DigiBox CODI will automatically turn them off for the duration of the effect.

3.15.13 Message Effect Push

Format: `\ME\PU\message\direction\speed{\mode}\\`

Parameters: `message` = message index (1 - 99999999) to push on

`direction` = push direction (`L,R,U,D`):

`L` = left
`R` = right
`U` = up
`D` = down

`speed` = push speed (1 - 20)

`1` = slowest

`2` = fastest

`mode` = ease mode (`S,E,SE,`optional)

`S` = ease in at start of push
`E` = ease out at end of push
`SE` = ease at start and end of push

Return: none

Description:

The Push command causes the current message to slide off the display as a new message is sliding into place. The background of the message is not affected (moved) by this command, only the foreground moves. The optional mode field can be used to accelerate the push from a stop to the specified speed and/or decelerate the push from full speed to a stop.

NOTE

If the clock and/or timer are displayed, DigiBox CODI will automatically turn them off for the duration of this effect.

3.15.14 Message Effect Roll Setup

Format: `\ME\RS\data\\`

Parameters: `data` = text data to roll. Data can also contain:

- Font Select command `\Fn\\`
- System Cursor `\SC\0or1\+-X\+-Y\\`
  (`X` can be absolute, or relative {w/+ or - sign}; `Y` is ignored)
- Tab Typewriter `\TT\N\\`
- Color Select command `\Cn\\`
- Row Center command `\RC\\`
- System Wait command `\SW\n\\` where `n` is wait time (NTSC: 60/sec, PAL: 50/sec)
- Row Right command `\RR\\` Note: `Y` value is ignored.
- Tab Position `\TP\N\\`

Return: none

Description:

The Roll Setup command configures the DigiBox CODI text and commands to be used during a roll. Data that can be used in the Roll Setup command are characters, plus a number of other commands as follows:

- Font Select (`\Fn\\`)
- Color Select (`\Cn\\`)
- Row Center commands (`\RC\\`)
- Row Right (`\RR\\`)
- System Wait (`\SW\n\\`)
- System Cursor (`\SC\\`)
- System Kern (`\SK\\`)
- Tab Position (`\TP\\`)
- Tab Typewriter (`\TT\\`)

The Row Center, Row Right and System Wait commands can appear anywhere on a roll data line in order to affect the line. Any other DigiBox CODI commands that are embedded in the roll data will generate an error response and will be ignored. The roll data can be up to 4500 characters. This command uses a portion of the available memory within the DigiBox CODI to save the roll data.

3.15.15 Message Effect Roll Begin

Format: `\ME\RB\speed\iteration{\yaddr\height}\\`

Parameters: `speed` = roll speed (0 - 9)

`0` = slowest
`9` = fastest

`iterations` = number of times to roll data, (0 = continuous, H = host data)

`yaddr` = y-pixel address to roll data (optional) (NTSC: 0 - 485, PAL: 10 - 575)

`height` = designates the extent of the "roll" function

Return: none

Description:

Once the roll data has been set up on the DigiBox CODI, the roll can begin. When the Roll Begin command is issued, the current display is shifted off the top of the screen as the roll data is shifted on from the bottom. The DigiBox CODI displays the complete roll for each iteration specified. If 0 iterations are specified, the roll continues until a new character or command is received from the host. DigiBox CODI will always complete the last iteration of the roll when a continuous roll is terminated. If the optional height (`H`) parameter is included with the command, the roll will be performed only in the area of the screen beginning and the `yaddr` and ending at a point represented by the value of `yaddr + height`.

If iterations is specified as 'H' (Host data), roll data is supplied directly from the host. The DigiBox CODI *does not* roll the Roll Setup data. (This mode does not require that the Roll Setup command be used before beginning the roll.) The host roll data can contain embedded commands as permitted in the Roll Setup command. The roll can be terminated by specifying the '\\' character string.

NOTE

- **If the roll data are supplied directly from the host, data previous set up using the Roll Setup command are erased. You must reset the Roll Setup data if it is required for a roll.**
- **The roll yaddr is an optional field and can be used to specify the start scanline of an active roll area of the display. The display above the roll y-pixel address remains static during the roll. If a yaddr is not specified, the roll uses the entire display.**
- **If the clock and/or timer are displayed, you must ensure that they are displayed within the static area. If the clock/timer are displayed within the active roll area, they will roll with the roll text.**

3.15.16 Message Effect Roll Erase

Format: `\ME\RE\\`

Parameters: none

Return: none

Description:

The Roll terminates after the last iteration of the roll is completed, or a character or command is received from the host in a continuous roll. If you do not plan to use the roll data again, you can erase the roll data using the Roll Erase command. This will free up memory allocated to the roll data which can then be used by other commands. Once the roll data has been erased using this command, you must set up a new roll using the Roll Setup command before a new roll can be performed.

3.15.17 Message Effect - iNFiNiT! Messages

Format: `\ME\I[infinit message filename]\effect.....\\`

Parameters: `I` = Identifies the message filename as an infinit message

`[iNFiNiT! message filename]` = filename of iNFiNiT! message

`effect` = Defines effect to be displayed

Return: none

Description:

iNFiNiT! /MAX/Maxine messages can now be read and displayed using DigiBox CODI effects. In all the Message Effects commands, the parameter that contains the message id can be formatted to specify an Infinit! Message. This is achieved by placing an '`I`' before the opening bracket for the full pathname or filename of the iNFiNiT! message to be read followed by the normal parameters for the required individual effect.

For example to format the *PUSH* effect, the following command can be used:

```
\ME\PU\I[c:\Infinit\Msg\.0001]\U\20\SE\\
```

This command will read in iNFiNiT! message .0001 and perform a Push Up effect at speed 20 using Start and Ending Ease.

NOTE

To use iNFiNiT! message effects, the iNFiNiT! Machine Font Directory that holds the iNFiNiT! machine fonts for these messages MUST have been previously set using the Set Font Directory command: `(\FP\[font directory path]\\).`

3.16 Palette Commands

The Download Palette command allows you to define the palette values for a character color using HSI or RGB values. The Color Setup command is used to set up the character color palette interactively.

The Upload Palette command allows you to upload the character palette values to the host. The host can save the uploaded palette values for future use. The following palette commands are described in this paragraph:

- **Download Palette (HSI)** **(§ 3.16.1, on 3-115)**
- **Download Palette (RGB)** **(§ 3.16.2, on 3-116)**
- **Upload Palette** **(§ 3.16.3, on 3-117)**
- **Download iNFiNiT! Palette** **(§ 3.16.4, on 3-118)**

3.16.1 Download Palette (HSI)

Format: `\DM\color\H\hbody\sbody\ibody\hedge\sedge\iedge {\color\H\hbody\sbody\ibody\hedge\sedge\iedge...}\\`

Parameters: `color` = color index (1 - 8) to set

`hbody` = hue body (0 - 360)

`sbody` = saturation body (0 - 255)

`ibody` = intensity body (0 - 255)

`hedge` = hue edge (0 - 360)

`sedge` = saturation edge (0 - 255)

`iedge` = intensity edge (0 - 255)

Return: none

Description:

The HSI Download Palette command is used to define the palette values for one or more colors. The command defines the hue, saturation and intensity color values for the character body and edge.

3.16.2 Download Palette (RGB)

Format: `\DP\color\R\rbody\gbody\bbody\redge\gedge\bedge{\color\R\rbody\gbody\bbody\redge\gedge\bedge...}\\`

Parameters:

`color` = color index (1 - 8) to set
`rbody` = red body (0 - 255)
`gbody` = green body (0 - 255)
`bbody` = blue body (0 - 255)
`redge` = red edge (0 - 255)
`gedge` = green edge (0 - 255)
`bedge` = blue edge (0 - 255)

Return: none

Description:

The RGB Download Palette command is used to define the palette values for one or more colors. The command defines the red, green and blue color values for the character body and edge.

3.16.3 Upload Palette

Format: `\UP\mode\\`

Parameters: **`mode`** = upload palette mode (1 - 9):
    `1-8` = upload palette for this color index
    `9` = upload palette for all colors

Return: `\DP\color\R\rbody\gbody\bbody\redge\gedge\bedge\\`

`color` = color index (1 - 8)

`rbody` = red body (0 - 255)\

`gbody`= green body (0 - 255)

`bbody` = blue body (0 - 255)

`redge` = red edge (0 - 255)

`gedge` = green edge (0 - 255)

`bedge` = blue edge (0 - 255)

Description:

The Upload Palette command sends the palette RGB information for a specified color to the host. The system provides both the character body and edge palette information for the specified color. By specifying color nine (9), the system responds with the palette information for all eight system colors.

3.16.4 Download iNFiNiT! Palette

Format: `\DP\I\\{palette data}`

Parameters:

Return: none

Description:

This function transfers an Infinit! Palette along with all its sub-palette information into the Digi-Box CODI. The command **`\DP\I\\`** is followed by 256 RGBA entries that make up the colors used in that palette. The Alpha value is always 0. The format of each palette entry is:

- **Byte 0: Red**
- **Byte 1: 0**
- **Byte 2: Blue**
- **Byte 3: Green**

Therefore, if it is necessary to load a 32 bit value into the palette array the following would be entered:

• **Full Red:**	`0x000000ff`
• **Full Blue:**	`0x00ff0000`
• **Full Green:**	`0xff000000`

The Palette Table is formatted in 8 rows by 32 columns. Each row represents a color group associated with a color index (C1-C8, C1=row 0, C8=row 7).

Each column represents a sub-palette entry. Entry 0 is the Face Color and entry 1 is the Edge in a simple font. Entries 2 to 31 are the sub-palette colors for multi-color characters.

3.17 Page Commands

The Page commands provide several editing functions which operate on the current display. Page commands affect an entire page or screen.

The following page commands are described in this paragraph:

- **Page Blank** **(§ 3.17.1, on 3-120)**
- **Page Center** **(§ 3.17.2, on 3-121)**
- **Page Erase** **(§ 3.17.3, on 3-122)**
- **Page Up** **(§ 3.17.4, on 3-123)**
- **Page Down** **(§ 3.17.5, on 3-124)**
- **Page Left** **(§ 3.17.6, on 3-125)**
- **Page Right** **(§ 3.17.7, on 3-126)**

3.17.1 Page Blank

Format: `\PB{\N}\\`

Parameters: **N** = erase the non-display buffer.

NOTE

The optional parameter \N can be used to erase the non-display buffer, while not affecting the background display. (See page erase command, § 3.17.3, on 3-122).

Return: none

Description:

The Page Blank command erases all characters on the page and positions the cursor at the Home position.

3.17.2 Page Center

Format: `\PC{\direction}\\`

Parameters: `direction` = page center direction (`V,H`, optional):

`V` = center page vertically only

`H` = center page horizontally only

Return: none

Description:

The Page Center command centers all rows on the screen. When the direction parameter is not specified, this command centers the display both vertically and horizontally. When the direction parameter is specified, the page is centered in that direction only.

3.17.3 Page Erase

Format: `\PE{\frame}\\`

Parameters: `frame:`

`0` = Displaying Frame
`1` = Non-Displaying Frame

Return: none

Description:

The Page Erase command erases both the background buffer and the foreground buffer at the same time.

3.17.4 Page Up

Format: `\PU{\scanlines}\\` <u>or</u>

`\PU{\scanlines}\N\\`

Parameters: `scanlines` = scanlines to shift up (optional)

`N` = Non-displaying buffer

Return: none

Description:

The Page Up command shifts all characters on the page up by a specified number of scanlines. The scanline parameter is optional. If the scanline count is not specified, the command moves the page up one scanline. If the **N** parameter is not present, shifting takes place in the displaying buffer. If the **N** parameter is present, shifting takes place in the non-displaying buffer.

3.17.5 Page Down

Format: `\PD{\scanlines}\\` or

`\PD{\scanlines}\N\\`

Parameters: **`scanlines`** = scanlines to push down (optional)

`N` = Non-displaying buffer

Return: none

Description:

The Page Down command shifts all characters on the page down by a specified number of scanlines. The scanline parameter is optional. If the scanline count is not specified, the command moves the page down one scanline. If the **N** parameter is not present, shifting takes place in the displaying buffer. If the **N** parameter is present, shifting takes place in the non-displaying buffer.

3.17.6 Page Left

Format: `\PL{\pixels}\\` <u>or</u>

`\PL{\pixels}\N\\`

Parameters: `pixels` = pixels to push left (optional)

`N` = Non-displaying buffer

Return: none

Description:

The Page Left command shifts all characters on the page to the left by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the page to the left by one pixel. If the **N** parameter is not present, shifting takes place in the displaying buffer. If the **N** parameter is present, shifting takes place in the non-displaying buffer.

3.17.7 Page Right

Format: `\PR{\pixels}\\` <u>or</u>

`\PR{\pixels}\N\\`

Parameters: **`pixels`** = pixels to push right (optional)

`N` = Non-displaying buffer

Return: none

Description:

The Page Right command shifts all characters on the page to the right by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the page to the right by one pixel. If the **`N`** parameter is not present, shifting takes place in the displaying buffer. If the **`N`** parameter is present, shifting takes place in the non-displaying buffer.

3.18 Row Commands

The Row commands provide several editing functions which operate on the current cursor row. DigiBox CODI provides up to 60 rows on the display at one time. The following commands are described in this paragraph:

- **Row Swap**	(§ 3.18.1, on 3-128)
- **Row Color**	(§ 3.18.2, on 3-129)
- **Row Font**	(§ 3.18.3, on 3-130)
- **Row Blank**	(§ 3.18.4, on 3-131)
- **Row Center**	(§ 3.18.5, on 3-132)
- **Row Up**	(§ 3.18.6, on 3-133)
- **Row Down**	(§ 3.18.7, on 3-134)
- **Row Left**	(§ 3.18.8, on 3-135)
- **Row Right**	(§ 3.18.9, on 3-136)
- **Row Insert**	(§ 3.18.10, on 3-137)
- **Row Expand**	(§ 3.18.11, on 3-138)
- **Row Squeeze**	(§ 3.18.12, on 3-139)

3.18.1 Row Swap

Format: `\RW\direction\\`

Parameters: `direction` = swap direction (`U,D`):

`U` = swap current row with row above

`D` = swap current row with row below

Return: none

Description:

The Row Swap command allows you to swap the position of the cursor row with either the row above or row below the cursor. After the rows are swapped, the cursor remains fixed on its original row position.

3.18.2 Row Color

Format: `\RO\\`

Parameters: none

Return: none

Description:

The Row Color command changes the character color(s) on the current row, starting with the cursor position. If the cursor is positioned at the first character in the row, the entire row is changed to the current color. Use the Color Select command to set the current color.

3.18.3 Row Font

Format: `\RF\\`

Parameters: none

Return: none

Description:

The Row Font command changes the character font(s) on the current row to the current font, starting with the cursor position. If the cursor is positioned at the first character in the row, the entire row is changed to the current font. Use the Font Select command to set the current font.

3.18.4 Row Blank

Format: `\RB\\`

Parameters: none

Return: none

Description:

The Row Blank command erases all characters on the cursor row and positions the cursor at the beginning of the row.

3.18.5 Row Center

Format: `\RC\\`

Parameters: none

Return: none

Description:

The Row Center command centers the characters on the cursor row. After the row is centered, the cursor is repositioned at the start of the row.

3.18.6 Row Up

Format: `\RU{\pixels}\\`

Parameters: `pixels` = pixels to push row up (optional)

Return: none

Description:

The Row Up command moves all characters on the cursor row up by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the row up one pixel.

3.18.7 Row Down

Format: `\RD{\pixels}\\`

Parameters: `pixels` = pixels to push row down (optional)

Return: none

Description:

The Row Down command moves all characters on the cursor row down by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the row down one pixel.

3.18.8 Row Left

Format: `\RL{\pixels}\\`

Parameters: `pixels` = pixels to push row left (optional)

Return: none

Description:

The Row Left command moves all characters on the cursor row to the left by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the row one pixel to the left. If the new position of a character is left of the screen edge, it is repositioned to the left-most edge.

3.18.9 Row Right

Format: `\RR{\pixels}\\`

Parameters: `pixels` = pixels to push row right (optional)

Return: none

Description:

The Row Right command moves all characters on the cursor row to the right by a specified number of pixels. The pixel parameter is optional. If the pixel count is not specified, the command moves the row one pixel to the right. If the right edge is closer than the specified number of pixels, DigiBox CODI will ignore the command.

3.18.10 Row Insert

Format: `\RI\\`

Parameters: none

Return: none

Description:

The Row Insert command allows insertion of a new row at the cursor row. The cursor row and any subsequent rows are moved down one row on the screen. The inserted row is blank.

3.18.11 Row Expand

Format: `\RE{\repeat}\\`

Parameters: `repeat` = iterations for `\RE\\` command (optional)

Return: none

Description:

The Row Expand command expands the spacing between all characters on the row. The repeat parameter is optional. If the repeat count is not specified, the command expands the character spacing by one pixel.

3.18.12 Row Squeeze

Format: `\RS{\repeat}\\`

Parameters: `repeat` = iterations for `\RS\\` command (optional)

Return: none

Description:

The Row Squeeze command squeezes the spacing between all characters on the row. The repeat parameter is optional. If the repeat count is not specified, the command squeezes the character spacing by one pixel.

3.19 System Commands

The System commands provide various DigiBox CODI functions for setting system operating modes and determining the DigiBox CODI operating status. The system commands that are described in this paragraph are listed below:

System Acknowledge	**(§ 3.19.1, on 3-141)**
System Cursor	**(§ 3.19.2, on 3-142)**
System Display	**(§ 3.19.3, on 3-143)**
System Execute	**(§ 3.19.4, on 3-144)**
System Home	**(§ 3.19.5, on 3-145)**
System Kern	**(§ 3.19.6, on 3-146)**
System Lock	**(§ 3.19.7, on 3-147)**
System Communication Mode	**(§ 3.19.8, on 3-148)**
System Overwrite/Insert	**(§ 3.19.9, on 3-150)**
System Patterns	**(§ 3.19.10, on 3-151)**
System Reset	**(§ 3.19.11, on 3-152)**
System Safe Title	**(§ 3.19.12, on 3-153)**
System Update	**(§ 3.19.13, on 3-155)**
System Wait	**(§ 3.19.14, on 3-156)**
Set 525 NTSC Display Mode	**(§ 3.19.15, on 3-157)**
Set 625 PAL Display Mode	**(§ 3.19.16, on 3-158)**
Display ON/OFF	**(§ 3.19.17, on 3-159)**

3.19.1 System Acknowledge

Format: `\SA\\`

Parameters: none

Return: `*<CR><LF>`

Description:

The System Acknowledge command is used when the Acknowledge mode has not been set and you want to know when a command has executed. To use the System Acknowledge command, send a DigiBox CODI command followed by the System Acknowledge command. Once the DigiBox CODI command has executed, the System Acknowledge command is processed and the acknowledge character is sent back to the host.

3.19.2 System Cursor (Prompt Channel)

Format: `\SC\mode\Xaddr\Yaddr\\`

Parameters:

`mode` = cursor display mode (`0` = off (default)

`xaddr` = x-pixel address of cursor (0 - 719)

`yaddr` = y-pixel address of cursor (NTSC: 0 - 485, PAL 0 - 575)

Return: none

Description:

The System Cursor command is used to enable or disable the cursor, select the cursor type, and specify its location. If the cursor is enabled, it will be visible on the Prompt Channel output. By default, the cursor is off.

Optionally, the System Cursor command allows you to position the DigiBox CODI cursor to any x,y coordinate on the display. When the xaddr or yaddr is not preceded by a plus (+) or minus (-) sign, the values are absolute. When a sign is used, the cursor is positioned relative to the current cursor position. If a character is displayed at the x,y specified, the system automatically aligns to the character.

3.19.3 System Display

Format: `\SD\1\workframe\\`

Parameters: `workframe` = work frame number (1, 2)

Return: none

Description:

DigiBox CODI contains two frame buffers. Frame 1 is the displaying frame and frame 2 is the non-displaying frame. By using the command `\SD\1\2\\` the working frame buffer can be set to the non-display buffer. This causes all operations to affect the non-display buffer and to leave the display buffer unaffected. Once the non-display buffer is built, it can be displayed using any of the Message Effect commands (§ 3.15, on 3-94) with a `-1` in the message number parameter. Using the `\SD\1\1\\` command will restore graphic operations to the display buffer. The System Display command, therefore, allows you to set the work frames.

3.19.4 System Execute

Format: `\SE\mode\\`

Parameters: `mode` = system command execute mode (`0,1`):

- `0` = display characters (default)
- `1` = ignore characters

Return: none

Description:

The System Execute command allows you to set the execute mode. In mode 0, characters received from the host are displayed on screen. In mode 1, all characters received from the host are ignored. The default is mode 0.

3.19.5 System Home

Format: `\SH\\`

Parameters: none

Return: none

Description:

The System Home command places the system cursor in the Home position, or top left corner, of the screen.

3.19.6 System Kern

Format: `\SK\{+,-}amount\\`

Parameters: `amount` = number of pixels to add or subtract between all characters:

Return: none

Description:

The System Kern command is used to globally add or subtract white space between characters that are sent to DigiBox CODI. The amount can be either positive (pixels are added between characters) or negative (pixels are subtracted). To clear the system kern command, use a value of zero.

3.19.7 System Lock

Format: `\SL\mode\\`

Parameters: `mode` = lock mode (`0,1`):

`0` = unlocked

`1` = locked (default)

Return: none

Description:

The System Lock command allows you to set the lock mode. In the Locked mode, all Letter commands affect the character at the cursor position and all trailing characters on that row. In the Unlocked mode, all Letter commands affect only the character at the cursor. The system defaults to Locked mode.

3.19.8 System Communication Mode

Format: `\SM\cmode\\`

Parameters: `cmode` = Communication mode (0 - 7, Table 3-3):

Table 3-3: Communication Mode

Comm. Mode	Acknowledge	Checksum
0	No	No
1	Yes	No
2	No	Yes+
3	Yes	Yes
4	No	No
5	Yes	No
6	No	Yes
7	Yes	Yes

Return: none

Description:

The System Communication Mode command allows you to set the communications parameters of the DigiBox CODI. In the Acknowledge mode, the DigiBox CODI returns an acknowledge character (*) after each command is processed. In Checksum mode, the DigiBox CODI requires a 5-digit ASCII checksum for each command sent to it. In addition, the DigiBox CODI adds a checksum onto all data sent to the host.

NOTE

If the checksum mode is enabled, the 5-digit checksum value following each command sent *to the* DigiBox CODI is the summation of the ASCII values of each command character with the exception of the initial backslash (\) character. If it is determined that the checksum does not match the command data received, the command is ignored and a checksum error code 050 is returned to the host. The host can respond by re-transmitting the command. In addition, DigiBox CODI appends a 5-digit checksum value to all data sent *to the host*. The host can then verify that there is no error in the data it has received.

Some sample command/checksum values are shown in Table 3-4:

Table 3-4: Sample Command Checksum Values

Command	Checksum
`\SM\1\\`	00485
`\SM\6\\`	00490
`\VV\\`	00356
`\SS\\`	00350
`\SF\\`	00337
`\SR\\`	00349
`\UB\1\\`	00476

3.19.9 System Overwrite/Insert

Format: `\SO\mode\\`

Parameters: `mode` = overwrite/insert mode (0, 1):
`0` = insert
`1` = overwrite (default)

Return: none

Description:

The System Overwrite/Insert command allows selection of the Overwrite or Insert mode. In Overwrite mode, new characters overwrite existing characters. In Insert mode, new characters are inserted in front of the character at the current cursor position. The system defaults to Overwrite mode.

3.19.10 System DigiBox CODI Patterns

Format: `\SP\pattern\\`

Parameters: `pattern` = PC Codi test pattern number (1 - 22):

Return: none

Table 3-5: DigiBox CODI Test Patterns

Number	Pattern
1	Full Field Color Bars
2	Full Field Color Bars Separated
3	SMPTE Bars (simulated)
4	Convergence Pattern
5	Black and White Bars
6	Y----- Ramp, 100 IRE
7	Yellow-Mode Ramp, 100 IRE
8	Blue-Mode Ramp, 100 IRE
9	Modulated Pedestal
10	Reverse Blue Bars
11	Red Field
12	Blue Field
13	Field Square Wave
14	Window/Field Bar
15	10 Step Staircase
16	Production Color Bars
17	Dump Palette Using Bars (RWM)
18	Dump Palette Using Squares
19	B-Y, R-Y Test Pattern
20	Quickfill
21	Background Key Test
22	Video Key Test

Description:

The DigiBox CODI can display 22 different test patterns. Use the System Pattern command to display one of the patterns.

3.19.11 System Reset

Format: `\SR\\`

Parameters: none

Return: 000*

Description:

The System Reset command performs a reset of the DigiBox CODI system. All system parameters are reset to their startup (default) settings. All messages, roll, crawl and GPI buffers are erased. The DigiBox CODI responds by sending the startup message (000*) to the host.

3.19.12 System Safe Title

Format: `\SG\mode\Xmin\Ymin\Xmax\Ymax\\`

Parameters: `mode` = Safe Title function mode (0, 1, 2):

`0` = off (cancels Safe Title Display)

`1` = on (Safe Title on; restricts characters to safe title area)

`2` = on (Safe Title on, does not restrict character placement)

`Xmin, Ymin` = upper left corner of Safe Title area*

`Xmax, Ymax` = lower right corner of Safe Title area*

* default value for Safe Title area is 80% of active video area. Boundaries for Safe Title area need to be specified only when changing the default value.

Return: none

Description:

This command is used to control the DigiBox CODI l Safe Title display. The Safe Title is an area on the monitor that demonstrates whether or not the text positioned in an area will be visible on a home television screen. This means that the monitor used to view the video will not have to underscan. The default value for safe title is 80% of active video. The size and location of the safe title area can be modified as needed.

Normally, the cursor will wrap to the next row when the right edge of safe title is encountered.

Enable Safe Title

To enable the safe title box, the `\sg\1\\` command is used.

As the characters are added to the frame buffer, the cursor will wrap to the next row when the right edge of the safe title is hit. The row begins at the left edge of the Safe Title. When the bottom of the safe title is hit, the cursor will move to the top left corner of the safe title. Please note that the cursor can still be positioned anywhere in the frame with the cursor position command.

Disable Safe Title

To disable the safe title box, the `\sg\0\\` command is used.

Enable Safe Title, No Clip to Safe Title Margins

To enable this function, the `\sg\2\\` command is used.

Change Safe Title Location and Size

The Safe Title location and size can be changed very easily. The format of the change command is as follows:

```
\sg\1\X Min\Y MIN\X MAX\Y MAX\\
```

The X MIN and Y MIN values mark the upper left corner of safe title. The X MAX and Y MAX values mark the lower right corner of safe title. The values can be absolute or relative. For relative values, precede the number with a plus (+) or minus (-) sign to change the value by that amount.

3.19.13 System Update

Format: `\SU\mode\\`

Parameters: `mode` = tab clip mode (0, 1):

- `0` = do not clip tab field characters (default), adjust characters
- `1` = clip tab field characters, adjust characters

Return: none

Description:

The System Update command allows selection of the clip mode for updating the text within a tab field using the Tab Update command. If mode is set to 0, there is no character clipping. If the mode is 1, characters are clipped when the Tab Update command is used.

3.19.14 System Wait

Format: `\SW\time\\`

Parameters: `time` = time to wait (NTSC: 60 counts/second, PAL: 50 counts/second)

Return: none

Description:

The System Wait command causes the DigiBox CODI to do nothing for the specified period of time. This command is useful for ensuring that the DigiBox CODI waits for the specified time before performing the next command. A command of \SW\90\\ will cause a wait period of one and a half seconds in NTSC. For PAL, the time specified for the same period would be \75\\.

3.19.15 Set 525 NTSC Display Mode

Format: `\S5\\`

Parameters: none

Return: none

Description:

This command configures the DigiBox CODI board to operate in the NTSC 525 Scanline Mode. It also sets the default state to the NTSC mode. After this command is executed, the setting is stored in the DigiBox CODI board and is invoked every time the board is booted. There is no need to reload this command.

3.19.16 Set 625 PAL Display Mode

Format: `\S6\\`

Parameters: none

Return: none

Description:

This command configures the DigiBox CODI board to operate in the PAL 625 Scanline Mode. It also sets the default state to the PAL mode. After this command is executed, the setting is stored in the DigiBox CODI board and is invoked every time the board is booted. There is no need to reload this command.

3.19.17 Display ON/OFF

Format: `\CD\{0/1}\\`

Parameters: 0 = Turns OFF echoing of commands being sent to digital pcCODI.

1 = Turns ON echoing of commands being sent to digital pcCODI.

Return: none

Description:

This command is used to enable or disable echoing of the commands that are being sent to the DigiBox CODI. This function is used as a tool for diagnosing problems during development of a DigiBox CODI interface. Operational details of the DigiBox CODI board are displayed on the associated Command Window while the board is running. This information is provided by the internal processor on the DigiBox CODI board.

NOTE

Use of this feature may slow-down some command operations. It should only be used during the code development process.

3.20 Tab Commands

A tab defines a position on the display, the font and color used for text displayed at the position, and the text justification mode. You can define up to 201 tab positions per message. Whenever the cursor is positioned to a tab, the system automatically uses the font, color and justification of the tab.

A tab can be identified by a numeric value (0 - 200), or by a name (up to 11 characters). Tab names provide a means of describing the tab type or function.

A tab “field” is defined as the area starting at the tab position and ending with either the x-pixel end address on the same display row, or the right edge of the display. In the event that no other tab positions are defined after the specified tab, the right edge of the display is used as the right boundary of the tab.

The tab text can be displayed using left, center or right justification within the tab field. Using left justification, text is positioned with the first character starting at the left-most boundary of the tab field. Center justification centers the tab text within the tab field. Finally, right justification positions the tab text at the right boundary of the tab field.

In addition to left, center and right justifications, tab text can be numerically justified. Numeric justification is used to display numeric values in a tab field using the width of the widest digit of tab font. The tab commands that are described in this paragraph are listed below:

• **Download Tabs**	**(§ 3.20.1, on 3-161)**
• **Tab Update (no display)**	**(§ 3.20.2, on 3-162)**
• **Upload Tabs**	**(§ 3.20.3, on 3-163)**
• **Tab Position**	**(§ 3.20.4, on 3-164)**
• **Tab Update**	**(§ 3.20.5, on 3-165)**
• **Tab Erase**	**(§ 3.20.6, on 3-166)**
• **Tab Typewriter**	**(§ 3.20.7, on 3-167)**
• **Highlight Tabs**	**(§ 3.20.8, on 3-168)**

3.20.1 Download Tabs

Format: **`\DT\tab\xaddr{,xaddrend}\yaddr \color\font\justification{,ext update, upper case only}{\tab \xaddr{,xaddrend}\yaddr\color\font \justification...}\\`**

Parameters: **`tab`**= tab number (0 - 200) or tab name (11 chars)

`xaddr` = x-pixel address for tab (0 - 719)

`xaddrend` = x-pixel end address for tab (0 - 719), optional

`yaddr` = y-pixel address for tab (NTSC: 0 - 485,
PAL: 0 - 575)

`color` = color index (1 - 8)

`font` = font index (1 - 16)

`justification` = text justification (**`L,LN,R,RN,C,CN`**):

`L` = left
`LN` = left numeric
`R` = right
`RN` = right numeric
`C` = center
`CN` = center numeric

`ext update, upper case only` = external flags, options to **`justification`** field:

`1` = enabled
`2` = disabled

Return: none

Description:

You can define up to 201 tab positions on the DigiBox CODI display. A tab is defined by a number or name, x,y pixel address, color, font and justification. Optionally, you can specify the end x-pixel address of the tab field. When not specified, the end of the tab field is the next tab or right edge of the display.

The Download Tab command allows you to define one or more tab positions on the display. When using numeric justification, all characters are displayed using the width of the widest digit of the font selected. A non-digit character displayed in a numeric justified field will also use the widest digit width unless the character width is larger.

Two optional flags are provided for the **`justification`** field. These are the **`external update`** flag and the **`upper case only`** flag. Both of these flags must be supplied if either is to be set. When set, the **`upper case only`** flag forces all codes in the ASCII range of a - z to be converted to A - Z. The **`external update`** flag will be passed back to the host during the Upload Tab command for the host to use in its Tab Updating operations.

3.20.2 Tab Update No Display

Format: `\TN\tab\data\{tab\data...}\\`

Parameters: `tab` = tab number (0 - 200)

`data` = new tab field text (max. 180 characters)

Return: none

Description:

The Tab Update No Display command updates the tab fields specified but does not automatically display the page when executed. This command is intended to be used with a display effect command that will display the updated information.

When the Tab Update command is used, any number of tab fields can be assigned.

This command has the same effect as first placing the system in the non-display buffer mode (`\SD\1\2\\`) and the executing a normal tab update command (`\TU....\\`).

3.20.3 Upload Tabs

Format: `\UT\mode\\`

Parameters: `mode` = upload mode (1, 2 or 3)

`1` = ASCII upload
`2` = include external update flags
`3` = return the ASCII text or hexadecimal Unicode contents of the tab field

Return:

```
\DT\tab \xaddr {,xaddrend} \yaddr\color \font \justification
{\tab \xaddr {,xaddrend} \yaddr \color \font
\justification...}\\
```

`tab` = tab number (0 - 200) or tab name (11 chars)

`xaddr` = x-pixel address of tab (0 - 719)

`xaddrend` = x-pixel end address for tab (0 - 719, optional)

`yaddr` = y-pixel address of tab (NTSC: 0 - 485, PAL 0-575)

`color` = color index (1 - 8)

`font` = font index (1 - 16)

`justification` = text justification (`L,LN,R,RN,C,CN`):

`L` = left
`LN` = left numeric
`R` = right
`RN` = right numeric
`C` = center
`CN` = center numeric

Description:

The Upload Tabs command sends current tab information for all defined tabs to the host. If set to `mode 2`, the Upload Tabs will cause the uploaded tab data to include the two extension fields of `external update` flag and `upper case only` flag. These fields are added to the `Justification` field in the Download Tabs command separated by commas.

If set to `mode 3`, the text string returned looks like the Tab Update command. For example:

`\TU\1\Hello From Tab 1\\` or

`\TU\1\[HUC030\C065\C078]\\`

3.20.4 Tab Position

Format: `\TP\tab\\`

Parameters: **`tab`** = tab number (0 - 200) or tab name (11 chars)

Return: none

Description:

The Tab Position command positions the cursor on the specified tab field. In addition, the system automatically sets the color and font index to what is defined for the tab field. Any text that was displayed at the tab field is erased.

Note that the <TAB> key (ASCII 9) can be used to advance the cursor, in sequential order, from one tab field to the next. Please note that the text currently displayed at the new tab field will not be cleared.

3.20.5 Tab Update

Format: **`\TU\tab\{Cx}data\{Fx}data{\tab\data...}\\`**

or

`\TN\tab\{Cx}data\{Fx}data{\tab\data...}\\`

Parameters: **`tab`** = Tab number (0 - 200) or tab name (11 chars)

`N` = Non-displaying buffer

`Cx` = Color override (1 - 8)

`Fx` = Font override (1 - 16)

`data` = Text to display at tab (max. 63 characters)

Return: none

Description:

The Tab Update command is used to update the text displayed at the specified tab. When a Tab Update command is used, you can specify any number of tab fields to be updated. To provide an immediate update of the display for all tab fields, DigiBox CODI builds the new screen in the non-displaying frame buffer. When all fields have been updated, the display is completely updated showing all updated tab fields.

The color of the tab data field can be changed by the **`Cx`** parameter, where **`x`** represents the color index (1 - 8). If the optional **`Cx`** parameter is used, it will override the color for the tab field for the characters that follow. The set color of the tab field will not change, and any future tab updates to that field will display in the original tab field color unless it is again overwritten by the **`Cx`** command. For example, in the command,

```
\TN\1\Hello[C2]there\\
```

the color of the word **`there`** would be changed to the color represented by color index `C2`.

If working buffer is set to the non-display mode, as represented by the **N** parameter, then the on-screen display will not change until a Message Effect (§ 3.15, on 3-94) command is executed to display it.

3.20.6 Tab Erase

Format: `\TE\tab{\tab...}\\`

Parameters: `tab`= tab number (0 - 200) or tab name (11 chars)

Return: none

Description:

The Tab Erase command is used to erase one or more tab definitions. This command affects only the tabs specified. If no **`tab`** parameter is specified (**`\TE\\`**), all TAB fields are cleared.

3.20.7 Tab Typewriter

Format: `\TT\tabnumber\\`

Parameters: `number` = Tab position to use

Return: none

Description:

The Tab Typewriter command positions the cursor at the X location of the specified tab number. The cursor is set to the color and font specified for that tab. The Y value of the tab field is not used.

3.20.8 Highlight Tabs

Format: `\TH\\`

Parameters:

Return: none

Description:

This function is used to draw a red box around all the Tab locations on the screen. This function is intended to be used for debugging and diagnostic purposes only.

3.21 Video Commands

The video commands that are described in this paragraph are listed below:

- **Video Transparency** (**§ 3.21.1, on 3-170**)
- **Video Window** (**§ 3.21.2, on 3-171**)
- **Video Blank** (**§ 3.21.3, on 3-172**)
- **Video Insert** (**§ 3.21.4, on 3-173**)
- **Video Only** (**§ 3.21.5, on 3-174**)
- **Video Verify** (**§ 3.21.6, on 3-175**)
- **Video Priority Set** (**§ 3.21.7, on 3-176**)
- **Video Exchange** (**§ 3.21.8, on 3-177**)
- **Force Video Key IN "OFF"** (**§ 3.21.9, on 3-178**)
- **Horizontal Video Adjustment** (**§ 3.21.10, on 3-179**)
- **Vertical Video Adjustment** (**§ 3.21.11, on 3-180**)
- **Video Genlock** (**§ 3.21.12, on 3-181**)
- **Video Capture Setup** (**§ 3.21.13, on 3-182**)
- **Video Capture Begin** (**§ 3.21.14, on 3-183**)
- **Video Capture Display** (**§ 3.21.15, on 3-184**)
- **Video Preview Display** (**§ 3.21.16, on 3-185**)

3.21.1 Video Transparency

Format: `\VT\percent\\`

Parameters: `percent` = percentage of background transparency (0 - 100)

`0` = off
`100` = fully on

Return none

Description:

The Video Transparency command is used to set the DigiBox CODI generated background transparency.

3.21.2 Video Window

Format: `\VW\mode\xaddr1\yaddr1\xaddr2\`
`yaddr2\\`

Parameters: `mode` = window mode (0 - 2):

- `0` = disable video window
- `1` = external video window on PC Codi background
- `2` = PC Codi background window on external video

`xaddr1` = start x-pixel address of window (0 - 719)

`yaddr1` = start y-pixel address of window (NTSC 0-485, PAL 0-575)

`xaddr2` = end x-pixel address of window (0 - 719)

`yaddr2` = end y-pixel address of window (NTSC 0-485, PAL 0-575)

Return: none

Description:

The Video Window command allows the creation of one of two types of window:

- an external video window on DigiBox CODI background
- a DigiBox CODI background window on external video.

The window can be positioned anywhere on the display and can be of any size. Use mode=0 to disable a previously-defined window.

3.21.3 Video Blank

Format: `\VB\{0/1}\`

Parameters: `0` = Disables Video Blank mode

`1` = Enables Video Blank mode

Return: none

Description:

The Video Blank command is used to enable or disable the use of external video supplied to the DigiBox CODI board. When the `1` function is set, the Video Blank mode is enabled and the incoming video that is applied to the DigiBox CODI is disabled. When the `0` function is set, the Video Blank mode is disabled and the incoming video signal that is applied to the board is enabled.

NOTE

The Video blank mode should be enabled (function `1` set) when the DigiBox CODI is being used in an upstream mode such as when the Video OUT and Key OUT signals are being used as inputs for insertion to a downstream device such as a switcher.

3.21.4 Video Insert

Format: `\VI\\`

Parameters: none

Return: none

Description:

The Video Insert command enables the DigiBox CODI keyer to add DigiBox CODI graphics and text to the video stream. This command performs the opposite function of the *Video Only Command* . After performing a video only function, where the DigiBox CODI graphics are removed from the display, the *Video Insert* command is enabled to replace the graphics.

3.21.5 Video Only

Format: `\vo\\`

Parameters: none

Return: none

Description:

The Video Only command allows the display of external video only, with DigiBox CODI-generated video blanked out. The Video Only command affects only the Digital (601) Video OUT. DigiBox CODI-generated characters and backgrounds will still be visible on the auxiliary panel Analog Monitor output while the program channel displays only external video. This allows DigiBox CODI generated video to be previewed on the analog monitor prior to insertion.

3.21.6 Video Verify

Format: `\VV\\`

Parameters: none

Return: none

Description:

The Video Verify command redraws the screen display.

3.21.7 Video Priority Set

Format: `\VP{\B/F}\\`

Parameters:
B = Background layer
F = Foreground layer

Return: none

Description:

When executed, this command toggles the DigiBox CODI display priority between the background and foreground layers. If the **B** parameter is entered, the DigiBox CODI background layer has priority over the foreground layer. Conversely, if the **F** parameter is entered, the foreground layer has priority over the background layer.

3.21.8 Video Exchange

Format: `\VX{\0/1\}\\`

Parameters: `1` = Sets video exchange from the Tab Update Mode

`0` = Disables video exchange from the Tab Update Mode

Return: none

Description:

The video exchange command controls how and when the both the Background and Foreground non-displaying buffers will be displayed. This command is useful when the system work buffer is set to the non-displaying buffer by the **`\SD\1\2\\`** command (§ 3.19.3, on 3-143).

When the **`\VX\\`** command ins issued, without an optional parameter, a vertical interval swap of the foreground and background frame buffers is performed. If the **`1`** parameter is included (**`\VX\1\\`**), then a video exchange is set from the Tab Update Mode. If the **`0`** parameter is included (**`\VX\0\\`**), video exchange from the Tab Update Mode is disabled.

3.21.9 Force Video Key IN "OFF"

Format: `\VY\{0/1}\\`

Parameters: `1` = Sets the Video Key IN signal to "OFF"

`0` = Enables the Video Key IN signal

Return: none

Description:

If a Key IN source signal is not available at the KEY IN BNC connector of the DigiBox CODI board and it is required to use the DigiBox CODI in an upstream mode (supplying a Video/Key signal to a downstream mixer), the **`\VY\1\\`** should be issued. If a Key IN signal source is available and is being used, then the **`\VY\0\\`** command should be issued.

Forcing the Video Key IN to the OFF position produces a cleaner Key Out Signal when the DigiBox CODI is being used as a an upstream device.

3.21.10 Horizontal Video Adjustment

Format: `\VO\{+,-}X Adj\\`

Parameters: `X Adj` = Horizontal adjustment

`+,-` = Direction of horizontal adjustment

Return: none

Description:

If it is required to align the pcCODI video to a separate Genlock Source, the **`\V0\{+,-}X Adj\\`** command is used.This command allows adjustment of the Horizontal reference of the DigiBox CODI video to the Genlock source in a positive (**`+`**) or negative (**`-`**) direction. The valid adjustment range is between -511 and +511.

3.21.11 Vertical Video Adjustment

Format: `\V1\Y Adj\\`

Parameters: `Y Adj` = Vertical adjustment

Return: none

Description:

If it is required to align the vertical reference of the pcCODI video to a separate Genlock Source, the **`\V1\Y Adj\\`** command is used. The valid adjustment range is between 0 - 7 scanlines.

3.21.12 Video Genlock

Format: `\VG\{A/D}\\`

Parameters: **`A`** = Analog Genlock

`D` = Digital Genlock

Return: none

Description:

This command sets the Genlock mode of operation, analog or digital. If **`A`** is set, Genlock is performed via the Analog Genlock IN BNC connector. If **`D`** is set, then Genlock if performed via the Digital VIDEO IN connector.

After this command is executed, the setting is stored in the DigiBox CODI board and is invoked every time the board is booted. There is no need to reload this command.

3.21.13 Video Capture Setup

Format: `\VZ\S\nSampleFrames\nFrameDelay\nDisplaySamples\\`

Parameters:

`nSampleFrames` = number of frames to capture (1 - 8)

`nFrameDelay` = Number of frames between each capture (1 - 120)

`nDisplaySamples`:

`1` = enable display of thumbnail samples

`0` = disable display of thumbnail samples

Return: none

Description:

The Video Capture Setup command can be used to capture up to eight successive frames of video. A delay of **`n`** frames can be inserted between each frame up to a total delay of 120 frames. The captured frames can be displayed as thumbnails on the display after the capture is done. Display of the captured frames is controlled by the **`nDisplaySamples`** parameter.

3.21.14 Video Capture Begin

Format: `\VZ\B\nSampleFrame\\`

Parameters: **`nSampleFrame`** = sets frames to be captured (0 ,1 - 8)

Return: none

Description:

Once the Video Capture Setup command has been issued to set the capture operation, the capture is started with the Video Capture Begin command. If all the frames defined in the Capture Setup command are to be captured, the **`nSampleFrame`** parameter must be set to **`0`**. The **`nSampleFrame`** parameter can be set to any number from 1 to 8 and a single frame capture will occur in that sample.

3.21.15 Video Capture Display

Format: `\VZ\F\nSampleFrame\\`

Parameters: `nSampleFrame` = sets captured frames to be displayed `(0 ,1 - 8)`

`1-8` = Display full captured screen

`0` = Display thumbnail samples

Return: none

Description:

Once a video capture operation has been completed, the Video Capture Display command can be used to display the full screen capture of the frame or the thumbnail samples. This is controlled by the **`nSampleFrame`** parameter.

NOTE

To transfer the captured frame to your application you must first set the full screen display. After that is done, use the Upload Display (`\UD`) command to set the parameter to `1` for a *.bmp* file. You can also use the `BltPixels()` function and set the `iDirection` parameter to `1`.

3.21.16 Video Preview Display

Format: `\VD\{1|0}\\`

Parameters:

`1` = Video preview display ON

`2` = Video preview display OFF

Return: none

Description:

This command is intended to be used for Video Capture when the DigiBox CODI is being used in upstream mode.

If the DigiBox CODI is to be used as an upstream device, meaning that the CODI Output Video and Key does NOT include incoming video, but pcCODI's video will be inserted by some other device downstream, this command will enable the Analog Monitor to display the incoming Video stream.

Normally when upstream mode is used the **Video Blank Command** is set to ON, `\VB\1\\`. This will prevent the DigiBox CODI from mixing the incoming Video and Key signals with the DigiBox CODI graphics. In this manner, only the CODI graphics video and key signals will be sent through the Video and Key Out connectors. At the same time, the Video does not appear on the Analog Monitor Out connector. To make Video Capture easier, this command allows the Video Stream from the Video In connector to be displayed on the Analog Monitor. When this command is **ON,** the internal DigiBox CODI graphics planes to the Analog Monitor are disabled and only the video is displayed. When the Video Capture operation has been completed, the DigiBox CODI will automatically reset back to the Video Preview Display **OFF** mode, thus enabling the captured images to be seen on the graphics planes.

3.22 Watch Commands

DigiBox CODI contains a real-time clock which can be used to display the time. The DigiBox CODI clock is also referred to as a “watch”. All clock commands use the Watch prefix to describe the command.

The clock time can be reset using the Watch Time command. The Watch Setup command is used to select the display parameters of the clock. The clock can be displayed in three different formats. In addition, each field can be displayed using any font or color.

In addition to the three display formats, you can create your own display format by repositioning the clock fields to any location on the display. You can add your own titles to describe each of the displayed fields.

The watch commands that are described in this paragraph are listed below:

• **Watch Time**	**(§ 3.22.1, on 3-187)**
• **Watch Setup**	**(§ 3.22.2, on 3-188)**
• **Watch Color**	**(§ 3.22.3, on 3-189)**
• **Watch Font**	**(§ 3.22.4, on 3-190)**
• **Watch Position**	**(§ 3.22.5, on 3-191)**
• **Watch Display**	**(§ 3.22.6, on 3-192)**
• **Watch Erase**	**(§ 3.22.7, on 3-193)**
• **Watch Index**	**(§ 3.22.8, on 3-194)**

3.22.1 Watch Time

Format: `\WT\hhmmss\\` (format 1) or

`\WT\{+-}hhmmss\\` (format 2)

Parameters:

format 1:

hhmmss = watch time to set (24 hour mode):
hh = hour
mm = minute
ss = second

format 2:

hhmmss = watch time to set (24 hour mode):
hh = hour
mm = minute
ss = second
+- = enables the currently selected Watch clock to have its time be based on Watch clock 1 plus or minus the number of hours specified.

Return: none

Description:

The Watch Time command sets the time on the DigiBox CODI The time is always specified in a 24-hour format (military time). Therefore, 8 a.m. would be 080000, and 8 p.m. would be 200000.

The Watch Time command specifies the starting time in hours, minutes and seconds for each Watch clock using format 1. Using format 2 a plus (+) or a minus (-) symbol can be placed before the hours parameter to enable the currently selected Watch clock to have its time be based on Watch clock 1 plus or minus the number of hours specified. This enables precise synchronization between multiple clocks whose time difference is related to the respective time zones. Thus the first Watch clock, index 1, could be loaded with 2:30 pm by calling the WT command with the value **`\WT\143000\\`**. Then select Watch clock 2 with the **`\WI\2\\`** command and set the Watch clock time with an offset of negative 1 hour with the **`\WT\-010000\\`** command. This will cause the second clock to display 1:30pm.

NOTE

Only the hours parameter can be used for the offset. The minutes and seconds field must be set to 0.

3.22.2 Watch Setup

Format: `\WS\xaddr\yaddr\color\font\format`
`\mode\\`

Parameters: **`xaddr`** = x-pixel address for watch text (0 - 719)

`yaddr` = y-pixel address for watch text
(NTSC 0-485, PAL 0-575)

`color` = text color index (1 - 8)

`font` = text font index (1 - 8)

`format` = watch format (text)

`mode` = watch mode (0 - 2)

Watch Format: @HH:MM:SS
HH:MM
MM:SS
@ = any format preceded by the character **N** displays digits numerically justified.

Watch Mode = watch mode (0 - 2):
0 = 24 hour clock
1 = 12 hour clock
2 = 12 hour clock with a.m./p.m.

Return: none

Description:

The Watch Setup command is used to specify how the clock time is to be displayed. Before the clock can be displayed, the Watch Setup command must be executed. The clock digits are normally positioned within each field, depending on the digit widths.

If you precede the format text with an ‘N’, DigiBox CODI will cause the timer digits to be displayed in a numerically justified format.

NOTE

DigiBox CODI supports either a numeric or text value for the format field.

3.22.3 Watch Color

Format: `\WC\field\color{\field\color...}\\`

Parameters: `field` = field to effect (`H,M,S`)

`H` = hours
`M` = minutes
`S` = seconds

`color` = color index (1 - 8) for field

Return: none

Description:

The Watch Color command is used to select the color for a specified field. The command should be used after the Watch Setup command to change the color attribute of any field. Multiple fields can be changed with a single command.

3.22.4 Watch Font

Format: `\WF\field\font{\field\font...}\\`

Parameters: `field` = field to effect (**H,M,S**):

H = hours
M = minutes
S = seconds

`font` = font index (1 - 16) for field

Return: none

Description:

The Watch Font command is used to select the font for a specified field. The command should be used after the Watch Setup command in order to change the font attribute of any field. Multiple fields can be changed with a single command.

3.22.5 Watch Position

Format: `\WP\field\xaddr\yaddr{\field \xaddr\yaddr...}\\`

Parameters: `field` = field to effect (`H,M,S`):

`H` = hours
`M` = minutes
`S` = seconds

`xaddr` = x-pixel address to display field (0 - 719)
`yaddr` = y-pixel address to display field
(NTSC 0-485, PAL 0-575)

Return: none

Description:

The Watch Position command is used to set the x,y coordinates of a field on the display. This command should be used *after* the Watch Setup command to modify the x,y position of any field. If this command is used, the field digits will always be right-justified. Multiple fields can be changed with a single command.

3.22.6 Watch Display

Format: `\WD\\`

Parameters: none

Return: none

Description:

The Watch Display command displays the clock on-screen after the watch setup command has executed.

3.22.7 Watch Erase

Format: `\WE\\`

Parameters: none

Return: none

Description:

The watch erase command erases (removes) the clock from the display.

3.22.8 Watch Index

Format: `\WI\watch\\`

Parameters: `watch` = Any value from 1 - 8

Return: none

Description:

There are up to eight (8) Active Watch Clocks available on the DigiBox CODI. This command allows the selection of any of the eight clocks. When working with multiple clocks, this command should be issued before any of the other Watch commands.

3.23 Miscellaneous Commands

With the introduction of the DigiBox CODI, a number of new, miscellaneous commands have been added to the DigiBox CODI command set. The new commands relate directly to the DigiBox CODI board and the software that interfaces with the board. The miscellaneous commands described in this paragraph are listed below:

Boot pcCODI	**(§ 3.23.1, on 3-196)**
Close pcCODI	**(§ 3.23.2, on 3-197)**
Build CODI Font	**(§ 3.23.3, on 3-198)**
Get pcCODI Information	**(§ 3.23.4, on 3-201)**
Get Number of DigiBox CODI Boards	**(§ 3.23.5, on 3-202)**
Command Capture On/Off	**(§ 3.23.6, on 3-203)**
Convert Chyron RGBA File to DIB	**(§ 3.23.7, on 3-204)**
Minimize Window‘	**(§ 3.23.8, on 3-205)**

3.23.1 Boot pcCODI

Format: `Boot(board number, .out filename)`

Parameters:

`board number` = the index of the DigiBox CODI board in the system, starting at 0

`.out filename` = the path and filename of the DigiBox CODI embedded software. (pcCODI601.out)

Return: `1` = Boot successful

`2` = Boot failed

Description:

The Boot pcCODI command resets the DigiBox CODI board, downloads the embedded software to the board and then starts the processor.

If the boot procedure failed, this could be caused by a DigiBox CODI board not being installed in the PC or a problem within the DigiBox CODI board. It can also be caused by the system not being able to find the ***pcCodi601.out*** file. If this occurs, a text file named ***boot.log*** will be created in the directory which will provide additional information concerning the boot process.

3.23.2 Close pcCODI

Format: `Close(board number)`

Parameters: `board number` = the index of the DigiBox CODI board in the system starting at 0.

Return: none

Description:

To free up the Windows resources, it is necessary to invoke the Close pcCODI command if the DigiBox CODI is no longer used, but the PC is being used for other purposes. If the Close pcCODI function is not invoked, the Windows application will continue to open the DigiBox CODI during start-up.If this occurs, memory resources will eventually be exhausted and the PCI interface will no longer operate. The ***boot.log*** file in the directory is updated with information regarding the Close process.

3.23.3 Build Codi Font - Function Call

Format:

```
BuildFont(&LOGFONT, char *output file name, int CodePage,
char *character set, int ExtraKern, int ExtraLeading, int
EdgeType, int EdgeAngle, int EdgeSize, int EdgeSofteness,
unsigned short  *pAddedUnicode, unsigned char
*pAddedUnicodePages, int palFlag);
```

Parameters:

`LOGFONT` = pointer to the logical font structure as defined in the Microsoft Developers Studio:

```
typedef   struct  tagLOGFONT  [ /* lf */
 LONG 1fHeight;
 LONG 1fWidth;
 LONG 1fEscapement;
 LONG 1fOrientation;
 LONG 1fWeight;
 BYTE  1fItalic;
 BYTE  1fUnderline;
 BYTE  1fStrikeOut;
 BYTE  1fCharSet;
 BYTE  1fOutPrecision;
 BYTE  1fClipPrecision;
 BYTE  1fQuality;
 BYTE  1fPitchAndFamily;
 CHAR 1fFaceName {LF_FACESIZE} ;
 ] LOGFONT;
```

`output filename` = This entry represents the filename that will contain the DigiBox CODI machine font.

`int CodePage` = This entry determines the Windows Code Page that will be selected. If this value is 0 or 1252, Windows Latin 1 (ANSI) code page will be selected. The other available code pages are listed in Table 3-6.

Table 3-6: Windows Code Pages

Code Page No.	Code Page Name
1250	Windows Latin 2 (Central Europe)
1251	Windows Cyrillic (Slavic)
1252	Windows Latin 1 (ANSI)
1253	Windows Greek
1254	Windows Latin 5 (Turkish)
1255	Windows Hebrew
1256	Windows Arabic
1257	Windows Baltic Rim

`character set` = This entry is a string of letters that represent the characters from the font from which they are to be rendered. If the string is null, then all available characters from the font will be rendered up to a maximum of 256 characters.

`ExtraKern` = This entry allows the spacing between characters for the selected font to be changed. A positive value increases the space between characters. A negative value reduces the space between characters.

`ExtraLeading` = This entry allows the line spacing of the selected font to be adjusted. A positive value increases the line spacing. A negative value reduces the line spacing.

`EdgeType` = This entry defines the type of edge to be applied to characters (0 - 3):

`0` = no edge
`1` = border
`2` = drop shadow
`3` = offset

`EdgeAngle` = This entry is used to specify the direction of the angle for EdgeTypes 2 and 3. The direction begins at 0 (12 o'clock), and continues in a clockwise direction from that position. Range of values is 0 - 359.

`EdgeSize` =This entry sets the size of the edge to be applied to the font. Size is given in scanlines.

`EdgeSofteness` = This entry is a number, ranging from 0 to 10, that is used to soften the character edge. A value of 0 indicates no softness.

`unsigned short *pAddedUnicode` = This entry is a pointer to a table of up to 255 UniCode characters that can be added to the basic font character set. Each basic font character set can contain up to 255 characters based on the standard code pages defined in Windows. This entry allows for an additional 255 characters to be extracted from a UniCode TrueType font and to be added to the Codi font. If this entry is NULL, no additional characters are created.

NOTE

If a table is supplied, then each table entry is a UniCode value that can be added to the font creation process that can be used by referencing the exact UniCode character ID in the display string. This table can be of any length up to 255 with the last entry being 0x0000.

`unsigned char *pAddedUnicodePages` = If entire blocks of UniCode pages are desired, this table will contain entries for the high order UniCode values that will reference those pages. All 256 characters in the page will be converted and added to the pcCODI machine font. It is necessary that this table be defined in ascending order.

`int palFlag` = During NTSC operation, this flag is set to 0. During PAL operation, this flag is set to 1. This maintains the correct aspect ration when creating a CODI font for use with PAL systems.

Return: none

Description:

DigiBox CODI fonts can be built from the PC-resident TrueType fonts with a call from the application. The font will be created and then saved in a disk file where it can be downloaded to the DigiBox CODI board using the Download Font command.

3.23.4 Get pcCODI Information - Function Call

Format: **`GetCODIInfo(INT32 iBoardIdx, CODIINFO* pCodiInfo)`**

Parameters: **`INT32 iBoardIdx`** = DigiBox CODI board number (0 - 7)

`CODIINFO* pCodiInfo` = Pointer to CODI information structure.

```
typedef struct  codiInfoS
{
 int iCrawlXPos;
 short iCommandActive;
 short iScanRate;
 } CODIINFO;
```

`iCrawlXPos` = Current position of Crawl and Crawl Build in Host.

`iCommandActive` = When `1,` command is being processed; when `0`, command processing has been completed.

`iScanRate`:

`0` = Uninitialized
`525` = NTSC
`625` = PAL

Return: none

Description:

3.23.5 Get Number of DigiBox CODI Boards - Function Call

Format: `int ipcCODIBoards = GetNumpcCODI()`

Parameters:

Return: number of DigiBox CODI boards

Description:

This function will return the number of DigiBox CODI boards installed in the PC.

3.23.6 Command Capture On/Off

Format: `\CC\{1,0}\\`

Parameters: `1` = Command capture function **ON**

`0` = Command capture function **OFF**

Return: none

Description:

When the Command Capture command is set to **ON**, (`\CC\1\\`) all the commands sent to the DigiBox CODI are saved in a file called *DpcCODICmds.txt.* This function is useful during development of an application program.

NOTE

The *DpcCODICmds.txt* file should be in the directory you run your program from and it will always be appended to. It is the responsibility of the application to remove this file when finished otherwise it will continue to grow as long as the Command Capture feature is active.

3.23.7 Convert Chyron RGBA File to DIB - Function Call

Format: `- intChyronRGBAtoBMP(char *pFileName,DIBSECTION *dsBMP,int iOpFlag);`

Parameters:

Return: `0` will be returned if message is not a Chyron RGBA message

Description:

This command is a DLL function call that converts a Chyron RGBA message into a Device Independent Bitmap (DIB) using the Windows *DIBSECTION* structure. This function reads the file whose file name is passed and determines if it is a Chyron RGBA message. A `0` will be returned if it is not a Chyron RGBA message. If not a Chyron RGBA message, a number other than `0` will be returned.

It is necessary that the user create a DIBSECTION structure. The specification for this can be found in Microsoft Windows Developer's Studio. To read a Chyron RGBA message, an initial call must be made to **`ChyronRGBAtoBMP()`** with the file name, a pointer to the DIBSECTION and the **`iOpFlag`** set to `0`. When the **`iOpFlag`** is set to `0` the file is read, tested to determine if it is a Chyron RGBA message, and the dimension of the RGBA message is returned. The dimensions must then be used to allocate a bitmap buffer to hold the Chyron RGBA Bitmap. Then **`ChyronRGBAtoBMP()`** must be called again with the pointer to the allocated buffer loaded into the DISBSECTION parameter and the **`iOpFlag`** set to `1`. This function call will load the bitmap data from the Chyron RGBA file into allocated buffer. The application can then use this information for any purpose.

3.23.8 Minimize TriMedia Console Window - Function Call

Format: `MinimizeWindow();`

Parameters:

Return: none

Description:

This function can be called after the DigiBox CODI has been booted. It minimizes the TriMedia Console window and places it on the Windows Task Bar.

DigiBox CODI Commands

COMMAND	NAME	PAGE
`\BE\...\\`	Background Erase	5-15
`\BL\...\\`	Set Scanline of Background Buffer	5-16
`\CD\...\\`	Display ON/OFF	5-159
`\CM\...\\`	Color Map	5-20
`\Cn\\`	Color Select	5-19
`\CR\...\\`	Color Restore	5-18
`\CT\...\\`	Color Transparency	5-21
`\DB\...\\`	Download Background (File-Blend) Mode 3	5-10
`\DB\...\\`	Download Background (File-Cut) Mode 4	5-11
`\DB\...\\`	Download Background (HSI)	5-6
`\DB\...\\`	Download Background (HSI) Mode 2	5-9
`\DB\...\\`	Download Background (PCI-Blend) Modes 5,6,7	5-12
`\DB\...\\`	Download Background (PCI-Cut) Modes 8 - 11	5-13
`\DB\...\\`	Download Background (RGB) Mode 1	5-7
`\DB\...\\`	Download Background (RGB) Mode 2	5-8
`\DF\...\\`	Download Font	5-51
`\DG\...\\`	Download Graphic	5-65
`\DI\...\\`	Download iNFiNiT! Font	5-52
`\DM\...\\`	Download iNFiNiT! Message	5-86
`\DM\...\\`	Download iNFiNiT! Message in Non- Display Buffer	5-87
`\DM\...\\`	Download Message	5-85
`\DP\...\\`	Download iNFiNiT! Palette	5-118
`\DP\...\\`	Download Palette (HSI)	5-115
`\DP\...\\`	Download Palette (RGB)	5-116
`\DT\...\\`	Download Tabs	5-161
`\DX\...\\`	Download Pixels	5-67
`\EB\...\\`	Event Timer Begin	5-34
`\EC\...\\`	Event Timer Color	5-38

DigiBox CODI Commands (Continued)

COMMAND	NAME	PAGE
`\ED\...\\`	Event Timer Display	5-31
`\EE\...\\`	Event Timer Erase	5-36
`\EF\...\\`	Event Timer Font	5-39
`\EI\...\\`	Event Timer Index	5-30
`\EL\...\\`	Event Timer Laps	5-35
`\EP\...\\`	Event Timer Position	5-40
`\ES\...\\`	Event Timer Setup	5-32
`\ET\...\\`	Event Timer Time	5-37
`\FB\\`	Flash Begin	5-47
`\FE\...\\`	Flash End	5-49
`\FM\...\\`	Flash Multiple	5-48
`\Fn\\`	Font Select	5-53
`\FP\...\\`	Set Font Directory Path	5-43
`\FS\...\\`	Flash Setup	5-46
`\GB\...\\`	GPI Begin	5-58
`\GE\...\\`	Graphic Erase	5-68
`\GI\...\\`	GPI Input	5-62
`\GL\...\\`	Graphic Transparency	5-69
`\GO\...\\`	GPI Output	5-61
`\GP\...\\`	Set Graphic Directory Path	5-44
`\GR\...\\`	GPI Time Reset	5-60
`\GS\...\\`	GPI Setup	5-57
`\GT\...\\`	GPI Time	5-59
`\HU\...\\`	Hexadecimal UniCode Characters	5-77
`\LB\\`	Letter Blank	5-72
`\LD\...\\`	Letter Down	5-74
`\LL\...\\`	Letter Left	5-75
`\LR\...\\`	Letter Right	5-76
`\LU\...\\`	Letter Up	5-73
`\MC\...\\`	Message Control	5-89
`\MD\...\\`	Message Display	5-88

DigiBox CODI Commands (Continued)

COMMAND	NAME	PAGE
\ME\...\	Message Effect - iNFiNiT! Messages	5-113
\ME\...\	Message Effect Cover	5-95
\ME\...\	Message Effect Crawl Begin	5-103
\ME\...\	Message Effect Crawl Setup	5-102
\ME\...\	Message Effect Dissolve	5-106
\ME\...\	Message Effect Fade	5-105
\ME\...\	Message Effect Hide	5-96
\ME\...\	Message Effect Matrix Wipe	5-107
\ME\...\	Message Effect Peel	5-97
\ME\...\	Message Effect Pour	5-98
\ME\...\	Message Effect Push	5-109
\ME\...\	Message Effect Reveal	5-99
\ME\...\	Message Effect Roll Begin	5-111
\ME\...\	Message Effect Roll Erase	5-112
\ME\...\	Message Effect Roll Setup	5-110
\ME\...\	Message Effect Split	5-100
\ME\...\	Message Effect Weave	5-101
\MF\...\	Fade Rectangle	5-25
\MN\...\	Message Number	5-93
\MP\...\	Set Message Directory Path	5-42
\MV\...\	Dissolve Rectangle	5-26
\PB\...\	Page Blank	5-120
\PC\...\	Page Center	5-121
\PD\...\	Page Down	5-124
\PE\...\	Page Erase	5-122
\PL\...\	Page Left	5-125
\PR\...\	Page Right	5-126
\PU\...\	Page Up	5-123
\RB\	Row Blank	5-131
\RC\	Row Center	5-132

DigiBox CODI Commands (Continued)

COMMAND	NAME	PAGE
`\RD\...\\`	Row Down	5-134
`\RE\...\\`	Row Expand	5-138
`\RF\\`	Row Font	5-130
`\RI\\`	Row Insert	5-137
`\RL\...\\`	Row Left	5-135
`\RO\\`	Row Color	5-129
`\RR\...\\`	Row Right	5-136
`\RS\...\\`	Row Squeeze	5-139
`\RU\...\\`	Row Up	5-133
`\RW\...\\`	Row Swap	5-128
`\S5\\`	Set 525 NTSC Display Mode	5-157
`\S6\\`	Set 625 PAL Display Mode	5-158
`\SA\\`	System Acknowledge	5-141
`\SC\...\\`	System Cursor	5-142
`\SD\...\\`	System Display	5-143
`\SE\...\\`	System Execute	5-144
`\SF\\`	System Font	5-54
`\SG\...\\`	System Safe Title	5-153
`\SH\\`	System Home	5-145
`\SK\...\\`	System Kern	5-146
`\SL\...\\`	System Lock	5-147
`\SM\...\\`	System Communication Mode	5-148
`\SO\...\\`	System Overwrite/Insert	5-150
`\SP\...\\`	System Patterns	5-151
`\SR\\`	System Reset	5-152
`\SU\...\\`	System Update	5-155
`\SW\...\\`	System Wait	5-156
`\TE\...\\`	Tab Erase	5-166
`\TN\...\\`	Tab Update (no display)	5-162
`\TP\...\\`	Tab Position	5-164
`\TT\...\\`	Tab Typewriter	5-167

DigiBox CODI Commands (Continued)

COMMAND	NAME	PAGE
`\TU\...\\`	Tab Update	5-165
`\UB\...\\`	Upload Background (HSI) Mode 2	5-14
`\UD\...\\`	Upload Display	5-24
`\UD\...\\`	Upload Pixels	5-70
`\UM\...\\`	Upload Message	5-90
`\UP\...\\`	Upload Palette	5-117
`\UR\...\\`	Upload Transparency	5-22
`\UT\...\\`	Upload Tabs	5-163
`\V1\Y Adj\\`	Vertical Video Adjustment	5-180
`\VB\...\\`	Video Blank	5-172
`\VG\...\\`	Video Genlock	5-181
`\VI\\`	Video Insert	5-173
`\VO\...\\`	Horizontal Video Adjustment	5-179
`\VO\\`	Video Only	5-174
`\VP\...\\`	Video Priority Set	5-176
`\VT\...\\`	Video Transparency	5-170
`\VV\\`	Video Verify	5-175
`\VW\...\\`	Video Window	5-171
`\VX\...\\`	Video Exchange	5-177
`\VY\...\\`	Force Video Key IN “OFF”	5-178
`\WC\...\\`	Watch Color	5-189
`\WD\\`	Watch Display	5-192
`\WE\\`	Watch Erase	5-193
`\WF\...\\`	Watch Font	5-190
`\WI\...\\`	Watch Index	5-194
`\WP\...\\`	Watch Position	5-191
`\WS\...\\`	Watch Setup	5-188
`\WT\...\\`	Watch Time	5-187
`\XC\...\\`	Draw Color	5-79
`\XE\...\\`	Draw Edge	5-80

DigiBox CODI Commands (Continued)

COMMAND	NAME	PAGE
`\XL\...\\`	Draw Line (To)	5-81
`\XM\...\\`	Draw Move (To)	5-82
`\XS\...\\`	Draw Size	5-83
`BltPixels`	Bit Blitting - *Function Call*	5-27
`Boot(board number)`	Boot pcCODI - *Function Call*	5-196
`BuildFont`	Build CODI Font - *Function Call*	5-198
`Close(board number)`	Close pcCODI - *Function Call*	5-197
`GetCODIInfo`	Get pcCODI Information - *Function Call*	5-201
`RegisterGPIO`	Register GPIO Callback Procedure - *Function Call*	5-63
Convert Chyron RGBA File to DIB	Convert RGBA File to DIB - *Function Call*	5-204
`MinimizeWindow();`	Minimize TriMedia Console Window	5-205

www.chyron.com